Г. Ф. Конахович, Д. О. Прогонов, О. Ю. Пузиренко

КОМП'ЮТЕРНА СТЕГАНОГРАФІЧНА ОБРОБКА Й АНАЛІЗ МУЛЬТИМЕДІЙНИХ ДАНИХ

ПІДРУЧНИК

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Г. Ф. Конахович, Д. О. Прогонов, О. Ю. Пузиренко

КОМП'ЮТЕРНА СТЕГАНОГРАФІЧНА ОБРОБКА Й АНАЛІЗ МУЛЬТИМЕДІЙНИХ ДАНИХ

ПІДРУЧНИК

Київ
«Центр учбової літератури»
2018

УДК 004.[056.5+624+93]
К 338

Рекомендовано до друку
Вченою радою Навчально-наукового інституту аеронавігації
Національного авіаційного університету
(протокол № 4 від 18.12.2017 р.)

та Вченою радою Національного авіаційного університету
(протокол № 5 від 24.01.2018 р.)

Рецензенти:
Мачуський Є. А. — д-р техн. наук, професор (Національний технічний університет "Київський політехнічний інститут" ім. І. Сікорського);
Юдін О. К. — д-р техн. наук, професор (Національний авіаційний університет);
Толюпа С. В. — д. т. н., професор (Київський національний університет імені Тараса Шевченка).

К 338 **Комп'ютерна стеганографічна обробка й аналіз мультимедійних даних**: підручник. / Г. Ф. Конахович, Д. О. Прогонов, О. Ю. Пузиренко. – Київ: «Центр учбової літератури», 2018. – 558 с.

ISBN 978-617-673-741-4

Розглянуто теоретичні й практичні основи обробки мультимедійних даних з використанням комп'ютерної стеганографії, а також методи стеганографічного аналізу графічного контенту інфокомунікаційних систем. Наочно показано особливості використання універсальної математичної системи Mathcad у цілях стеганографічної обробки даних.

Представлено приклади практичної реалізації приховання даних у статичних зображеннях, аудіосигналах і тексті.

Системно викладені проблеми надійності та стійкості довільної стеганографічної системи по відношенню до різноманітних типів атак, а також оцінки пропускної здатності каналу прихованого обміну даними.

Представлені результати інформаційно-практичного дослідження проблеми стеганографічного аналізу цифрових зображень.

ISBN 978-617-673-741-4

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АК	ансамбль класифікаторів
АКФ	автокореляційна функція
АС	автоматизована система
АСА	активний стеганографічний аналіз
БК	базовий класифікатор
БЧХ	код Боуза-Чоудгурі-Хоквінгема
ВА	варіограмний аналіз
ВКФ	взаємокореляційна функція
ВЧ	високочастотний (сигнал)
ГПВП	генератор псевдовипадкової перестановки
ГПВФ	генератор псевдовипадкової функції
ДВП	дискретне вейвлет перетворення
ДКП	дискретне косинусне перетворення
ДМО	демаскуюча ознака
ДПФ	дискретне перетворення Фур'є
ЗСД	загальний стеганографічний детектор
ЗСЛ	зорова система людини
ІКМ	імпульсно-кодова модуляція
КВЗ	канал відкритого зв'язку
КЛР	класифікатор на основі лінійної регресії
КПЗ	канал прихованого зв'язку
КППД	канал передавання прихованих даних
КС	комп'ютерна стеганографія
ЛДФ	лінійний дискримінант Фішера
ЛРЗЗЗ	лінійний регістр зсуву зі зворотним зв'язком
МФА	мультифрактальний аналіз
МФС	мультифрактальний спектр
МФФА	мультифрактальний флуктуаційний аналіз
МЯ	метрика якості
НЗБ	найменший значущий біт
НЧ	низькочастотний (сигнал)
ОПЦЗК	область перетворення цифрового зображення-контейнера
ПА	перетворення Арнольда

ПВП	псевдовипадкова послідовність
ПВЧ	псевдовипадкове число
ПЗ	пропускна здатність
ПКЛ	перетворення Карунена-Лоева
ППЗ	прихована пропускна здатність
ПСА	пасивний стеганографічний аналіз
РГБ	розмірність Гаусдорфа-Безіковича
РС	розширення спектра сигналу
РСПП	розширення спектра сигналу прямою послідовністю
СА	стеганографічний алгоритм
СД	стеганографічні дані
СМ	стеганографічний метод
СМК	статистична модель контейнера
ССЛ	слухова система людини
СУЕГ	спектр узагальнених експонент Герста
СУФР	спектр узагальнених фрактальних розмірностей
СЧ	середньочастотний (сигнал)
УЕГ	узагальнені експоненти Герста
УСД	універсальний стеганографічний детектор
ФГ	фільтр Гауса
ЦВЗ	цифровий водяний знак
ЦЗ	цифрове зображення
ЦЗК	цифрове зображення-контейнер
ЦОС	цифрова обробка сигналів
ЦС	цифрова стеганографія
ШПФ	швидке перетворення Фур'є

ASCİİ	американський стандартний код для обміну інформацією (*American Standard Code for İnformation İnterchange*)
BD	формат оптичних дисків, що використовуються для зберігання відео високої чіткості і даних (*Blu-ray Disc*)
BMP	формат бітового відображення графічного об'єкта (*BitMaP*), в якому растрове зображення зберігається у вигляді двовимірного масиву пікселів
CR	службовий *ASCİİ*-код, що позначає операцію повернення курсору (каретки) — переведення його до лівого краю аркуша при виведенні тексту на символьний пристрій (*Carriage Return*)
CRC	контроль циклічним надмірним кодом (*Cyclic Redundancy Check*)
DCT	дискретне косинусне перетворення (*Discrete Cosine Transform*) — математичне перетворення, використовуване в алгоритмах компресії зображень (наприклад, у *JPEG*)
FDCT	пряме дискретне косинусне перетворення (*Forward Discrete Cosine Transform*)
GİF	формат обміну графічними даними (*Graphics İnterchange Format*), широко використовуваний для зберігання простих растрових зображень, що містять великі поля одного кольору
İDCT	зворотне дискретне косинусне перетворення (*İnverse Discrete Cosine Transform*)
İTU	Міжнародний союз електрозв'язку (*İnternational Telecommunication Union*)
JPEG	розроблений групою експертів з машинної обробки фотографічних зображень (*Joint Photographic Experts Group*) стандарт стиснення зі втратами повноколірних нерухомих зображень на основі алгоритму дискретного косинусного перетворення
LF	службовий *ASCİİ*-код, який викликає переведення курсору на екрані до тієї ж самої колонки на один рядок нижче (*Line Feed*)
LFSR	лінійний регістр зсуву зі зворотним зв'язком (*Linear Feedback Shift Register*)
LSB	молодший значущий біт (розряд) двійкового числа (*Least Significant Bit*)

MPEG	група стандартів на стиснення рухомих зображень і звуку (*Motion Picture Experts Group*)
MSB	старший значущий біт (розряд) двійкового числа (*Most Significant Bit*)
PCM	імпульсно-кодова модуляція (*Pulse Code Modulation*)
RGB	основна палітра: «червоний, зелений, синій» (*Red-Green-Blue*), що використовується у комп'ютерній графіці та програмуванні
SS	розширення спектра сигналів (*Spread Spectrum*)
VPN	віртуальна приватна мережа (*Virtual Private Network*), підмережа корпоративної мережі, яка забезпечує безпечне входження в неї віддалених користувачів
XOR	виключна диз'юнкція (нееквівалентність, додавання за модулем 2) (*eXclusive OR*) — бінарна логічна операція, результат якої є істинним лише тоді, коли значення операндів не збігаються

ВСТУП

Інформація є однією з найцінніших речей у сучасному житті. З появою глобальних комп'ютерних мереж отримання доступу до неї надзвичайно спростилося. Утім, за відсутності заходів щодо захисту даних, легкість і швидкість такого доступу значно підвищили й рівні таких загроз, як неавторизований доступ, фальсифікація, «піратство» тощо.

Задачі надійного захисту авторських прав, прав інтелектуальної власності або конфіденційних даних (які здебільшого мають цифровий формат) є одними з найстаріших і повністю не вирішених на сьогодні. У зв'язку з інтенсивним розвитком і поширенням технологій, які дозволяють за допомогою персонального комп'ютера інтегрувати, обробляти і синхронно відтворювати різноманітні типи сигналів (так звані *мультимедійні технології*), питання захисту інформації, представленої у цифровому форматі, є надзвичайно актуальними. Переваги представлення і передавання саме цифрових даних (простота відновлення, висока потенційна завадостійкість, перспективи використання універсальних апаратних і програмних рішень) можуть бути перекреслені з тією ж легкістю, з якою можливі їх викрадення і модифікація. Тому в усьому світі давно назрілим є питання розробки методів і заходів захисту інформації організаційного, методологічного і технічного характеру, основними серед яких є методи криптографії і стеганографії.

Криптографічний (з грецької *χρυπτός* — «таємний», *γράφω* — «пишу») захист інформації (система зміни останньої з метою зробити її незрозумілою для непосвячених, приховання змісту повідомлень за рахунок їх шифрування) не знімає зазначену вище проблему повністю, оскільки наявність шифрованого повідомлення сама по собі привертає увагу і зловмисник, заволодівши, наприклад, захищеним криптографічно файлом, маючи підозри про розміщення в ньому певної секретної інформації і за наявності належної зацікавленості цілком здатен скористатися наявним у його розпорядженні обчислювальним ресурсом для дешифрування даних.

Приховання ж самого факту існування секретних даних при їх передаванні, зберіганні чи обробці є задачею *стеганографії* (з грецької *στεγανός* — «прихований») — науки, що вивчає способи і методи приховання конфіденційних відомостей. Задача видобування інформації при цьому відступає на другий план і в більшості випадків розв'язується стандартними криптографічними методами. Інакше кажучи, під приховання існування розуміється не лише унеможлив-

лення виявлення в перехопленому повідомленні наявності іншого (прихованого) повідомлення, але й взагалі зробити неможливим викликання підозр стосовно цього, оскільки в останньому випадку проблема інформаційної безпеки повертається до стійкості криптографічного коду. Таким чином, займаючи свою нішу в забезпеченні інформаційної безпеки, стеганографія не замінює, а, скоріше, доповнює криптографію [1]. Процедура стеганографічного захисту може здійснюватися найрізноманітнішими способами, загальною рисою яких є те, що приховане повідомлення вбудовується в деякий не привабліючий увагу об'єкт (контейнер), який згодом відкрито транспортується (надсилається) адресатові.

Історично напрямок стеганографічного приховання інформації був першим [2], але згодом з багатьох причин він був витіснений криптографією. Інтерес до стеганографії відродився наприкінці XX ст. і був пов'язаний з масовим розповсюдженням технологій мультимедіа (що є цілком закономірним, з огляду на зазначені вище проблеми, пов'язані з захистом інформації). Не менш важливою стала і поява нових способів здійснення інформаційного обміну, що в сукупності з першим фактором надало нового імпульсу розвиткові та удосконаленню стеганографії, сприяло виникненню нових стеганографічних методів, в основу яких було закладено особливості представлення інформації в комп'ютерних файлах, обчислювальних мережах і т. п. Це, у свою чергу, дає можливість казати про існування окремого напрямку у сфері захисту інформації — *комп'ютерної стеганографії* (КС) [3–5, 19].

З 1996 р. проводяться міжнародні симпозіуми, присвячені проблемам приховання даних (*Information Workshop on Information Hiding*). Перша конференція, присвячена стеганографії, відбулася у липні 2002 р. На сьогодні стеганографія вже є наукою, яка продовжує швидко і динамічно розвиватися, використовуючи при цьому методи і досягнення криптографії, цифрової обробки сигналів, теорії зв'язку та інформації.

Приховане передавання даних (так звана *класична стеганографія*) є не єдиною сферою застосування КС. Методи стеганографії дозволяють успішно вирішувати задачі завадостійкої автентифікації, захисту інформації від несанкціонованого копіювання, відстеження шляхів поширення інформації мережами зв'язку, пошуку інформації в мультимедійних базах даних тощо. Усі ці обставини дозволяють в межах традиційно існуючих інформаційних потоків чи інформаційного середовища вирішувати і певні важливі питання захисту інформації низки прикладних галузей.

Існують два ключових напрямки використання КС: пов'язаний з цифровою обробкою сигналів (ЦОС) і не пов'язаний. У першому випадку секретні повідомлення вбудовуються у цифрові дані, які, як правило, мають аналогову природу свого походження (мова, зображення, аудіо- і відеозаписи) [1, 3–5]. У другому — конфіденційна інформація розміщується в заголовках файлів чи пакетів даних. Але цей напрямок не знайшов широкого застосування через відносну легкість розкриття і/або знищення прихованої інформації. Переважна більшість поточних досліджень в галузі стеганографії так або інакше пов'язана саме з ЦОС. Що, у свою чергу, дозволяє вже говорити про *цифрову стеганографію* (ЦС) [5, 19].

Можна виділити щонайменше дві причини популярності в наш час досліджень в галузі стеганографії: обмеження на використання криптографічних засобів в низці країн світу і поява проблеми захисту прав власності на інформацію, представлену у цифровому вигляді.

Перша причина спричинила велику кількість досліджень у дусі класичної стеганографії (тобто приховання власне факту здійснення передавання), друга — не менш чисельні роботи в галузі так званих цифрових водяних знаків (ЦВЗ) — спеціальних міток, приховано вбудовуваних до різноманітного мультимедійного контенту з метою можливості контролювання його використання.

Приховання інформації лише завдяки факту невідомості зловмиснику методу або методів, закладених до основи приховання, на сьогоднішній день є малоефективним. Ще у 1883 р. фламандський криптограф Огюст Керкгоффс *(A. Kerckhoffs)* писав про те, що система захисту інформації повинна виконувати покладені на неї функції навіть за повної поінформованості супротивника про її структуру і алгоритми функціонування [6]. Уся секретність системи захисту передаваних повідомлень має міститися лише в ключі — фрагменті інформації, що є попередньо (як правило) розділеним між адресатами. Незважаючи на те, що цей принцип відомий вже більше 100 років, і досі зустрічаються розробки, що ними зневажають. Очевидно, що вони можуть застосовуватися лише з навчальною метою.

В основі багатьох підходів до вирішення задач стеганографії лежить загальна з криптографією методична база, яку заклав ще всередині минулого століття Клод Шеннон *(C. E. Shannon)* [45, 60]. Але й дотепер теоретичні основи стеганографії є недостатньо висвітленими у сучасній літературі.

Беручі до уваги вищесказане, можна зробити висновок, що на сьогодні існує актуальна науково-технічна проблема удосконалення алгоритмів і методів проведення стеганографічного приховання конфіденційних даних або захисту авторських прав на певну

інформацію мультимедійного типу. На сьогодні абсолютно не бракує стеганографічних програм як початкового, так і професійного рівня (*S-Tools*, *Steganos Security Suite*, *bmpPacker* та ін.). Але захищеність їх коду (особливо це стосується програм професійного рівня) не дозволяє простежити методи, закладені в основу алгоритмів їх дії. Викладені ж на *Internet*-ресурсах численні тексти програм, через свою низьку інформативність для непрограмістів, мало чим зараджують, оскільки їх компіляція має своїм результатом вже виконувану програму з вкрай важко простежуваним алгоритмом дії, оскільки видає вже готовий результат — заповнений стеганоконтейнер, — заздалегідь встановити достатність рівня прихованості конфіденційної інформації в якому практично не можливо. Отже, є цілком очевидною нестача саме програм навчального (початкового) рівня, які б крок за кроком наочно демонстрували весь процес стеганографічного перетворення, що можна було б використати в учбовому процесі при підготовці фахівців у сфері захисту мультимедійного контенту в сучасних телекомунікаційних системах і мережах.

Стан порушеного питання у сфері стеганографії характеризується наступними основними досягненнями. Питання стеганографічного приховання секретних повідомлень, включно з побудовою ефективних алгоритмів приховання, свого часу розглядали у своїх роботах Ґуставус Сіммонс *(G. J. Simmons)*, Джесіка Фрідріх *(J. Fridrich)*, Рос Андерсон *(R. J. Anderson)*, Даніель Грул *(D. Gruhl)*, Норішіге Морімото *(N. Morimoto)*, Крістіан Кешін *(C. Cachin)*, Іоаніс Пітас *(I. Pitas)* та ін. [7–9, 13–16]. Результати досліджень стеганографічних алгоритмів на стійкість приводять у своїх працях *J. Fridrich*, Річард Попа *(R. Popa)*, Ніл Джонсон *(N. F. Johnson)*, Сушіль Джаджодіа *(S. Jajodia)*, Святослав Волошиновський *(S. Voloshynovskiy)* та ін. [9, 17, 18, 20, 40, 41]. Також необхідно відзначити праці авторів Біргіт Фіцманн *(B. Pfitzmann)*, Брюс Шнаєр *(B. Schneier)* і Скотт Крейвер *(S. Craver)* з питань узгодження термінології та формування основних стеганографічних протоколів [10–12].

Тривалий час у вітчизняній літературі і літературі країн СНД стеганографії було присвячено лише декілька оглядових журнальних статей [1, 4, 22–24, 48]. Крім того, заслуговують на увагу роботи [3] під авторством Володимира Хорошка, Олексія Азарова, Михайла Шелеста і Юрія Яремчука, а також [5] під авторством Вадима Грибуніна, Ігоря Окова та Ігоря Турінцева, заслугою яких є чи не перша спроба системного викладення стеганографічних методів, узагальнення найостанніших результатів досліджень у сфері комп'ютерної стеганографії. Вже згодом з'явилися більш вузькоспеціалізовані видання на зразок [109–111,123].

У 2006 р. вийшла друком і монографія [112] Георгія Конаховича і Олександра Пузиренка, метою якої було викладення теоретичних і, що не менш важливо в освітніх цілях, практичних основ комп'ютерної стеганографії, для чого були розглянуті особливості і перспективи використання в цілях стеганографічного захисту інформації вельми наочної і доступної для розуміння пересічними користувачами системи символьної математики *Mathcad*. Книга, що її ви наразі тримаєте у своїх руках, є другим, дещо переробленим і більш розгорнутим у плані стеганографічного аналізу виданням.

Проведено аналіз спеціалізованих літературних джерел та ресурсів мережі *Internet* щодо перспективних напрямків, за якими можливе використання стеганографії як інструменту захисту інформації в автоматизованих системах обробки даних. Шляхом дослідження відомих публікацій вітчизняних і закордонних авторів здійснено системне викладення проблем надійності і стійкості довільної стеганографічної системи по відношенню до видів здійснюваних на неї атак, а також оцінки пропускної здатності каналу прихованого обміну даними, яким, по суті, і є стеганографічна система. Наведено результати існуючих інформаційно-теоретичних досліджень проблеми інформаційного приховання у випадку активної протидії порушника.

Здійснено системне викладення відомих стеганографічних методів, спрямованих на приховання конфіденційних даних у комп'ютерних файлах графічного, звукового і текстового форматів.

Наведено приклади програмних комплексів для демонстрації принципів, закладених в основу методів стеганографічного приховання інформації у просторовій (часовій) або частотній областях використовуваного мультимедійного контейнера.

Використання під час комп'ютерного моделювання універсальної математичної системи *Mathcad* дозволяє використовувати потужні засоби реалізації чисельних методів розрахунку і математичного моделювання у поєднанні з можливістю виконання операцій символьної математики [25, 26]. Сторони, що здійснюють прихований обмін даними, практично позбавляються необхідності у програмуванні власне розв'язку задач, на них лише покладається коректний опис алгоритму розв'язку на вхідній мові *Mathcad*, що є мовою дуже високого рівня. Зазначене є суттєвою перевагою у порівнянні з існуючими на сьогодні додатками, написаними за допомогою вузькоспеціалізованих мов програмування на зразок *C/C++*, *Java* тощо. Останні, хоча і відрізняються більш високим рівнем гнучкості з точки зору можливостей реалізації тих або інших методів стеганографії, проте характеризуються незрівнянно тривалішим внесенням змін до вже написаної програми і наступної її компіляції. Час, затрачуваний на

внесення модифікацій, стає особливо важливим у випадку багатоетапних досліджень, що мають місце при використанні програми в навчальному процесі.

Завдяки своїй наочності та можливості швидкого проведення модифікацій програмних модулів, розроблені комплекси відповідають вимогам, що ставляться до програм, використовуваним у навчальних цілях. Підхід поєднання теоретичного викладення матеріалу з демонстрацією його практичного використання дозволяє позбавитися абстрактності формулювань, прийнятої у спеціалізованій і довідковій літературі з інформаційної безпеки, і сприяє розвитку у студентів здорового інтересу до практичних аспектів вирішення науково-технічних задач із захисту інформації. Книга може використовуватися в якості довідникового посібника з питань комп'ютерної стеганографії при використанні сучасних комп'ютерно-математичних систем.

Також одним з основних завдань даної книги є демонстрація ключових принципів, закладених в основу поширених на сьогодні методів стеганографічної обробки мультимедійних даних з можливістю проведення стеганоаналізу.

Книгу призначено для фахівців, які працюють у сфері захисту інформації і зацікавлені в ефективному використанні можливостей сучасних обчислювальних систем, а також для студентів і викладачів ВНЗ, які навчаються чи спеціалізуються у сфері інформаційної безпеки в телекомунікаційних системах і мережах.

Розділ 1. МІСЦЕ СТЕГАНОГРАФІЧНИХ СИСТЕМ У СФЕРІ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

1.1. Основні джерела й наслідки атак на інформацію, що обробляється в автоматизованих системах

Рівень розвитку й безпека інформаційного простору, що є системоутворюючими факторами в усіх сферах національної безпеки, активно впливають на стан політичної, економічної, оборонної та інших складових національної безпеки України.

Інформаційна безпека, захист якої відповідно до ст. 17 Конституції України, поряд із суверенітетом, територіальною цілісністю та економічною безпекою, є найважливішою функцією держави, досягається шляхом розробки сучасного законодавства, впровадження сучасних безпечних інформаційних технологій, побудовою функціонально повної національної інфраструктури, формуванням і розвитком інформаційних відносин тощо.

Зокрема, Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» [33] дає наступне визначення терміну *«захист інформації»* в інформаційних, телекомунікаційних та інформаційно-телекомунікаційних системах — це діяльність, спрямована на запобігання несанкціонованим діям щодо інформації в зазначених системах. Окремим видом захисту інформації є *технічний захист*, спрямований на забезпечення за допомогою інженерно-технічних заходів та/або програмних і технічних засобів унеможливлення витоку, знищення та блокування (унеможливлення доступу) інформації, порушення цілісності (зміни вмісту) та режиму доступу до інформації. *Доступом до інформації* є отримання користувачем (фізичною або юридичною особою) можливості обробляти інформацію в системі. Дії, що провадяться з порушенням порядку доступу до інформації, встановленого відповідно до законодавства, є *несанкціонованими*.

Неоднозначним, на думку фахівців, є трактування поняття *«атака на інформацію»*, оскільки остання, особливо у цифровому вигляді, може бути представлена сотнями різноманітних підвидів. Інформацією можна вважати й окремий файл, і базу даних, і лише один запис в ній, і цілий програмний комплекс. На сьогодні всі ці об'єкти зазнають або можуть зазнавати атакуючих дій з боку певної особи (або групи осіб) — *порушника*.

При зберіганні, підтримці та наданні доступу до будь-якого інформаційного об'єкта його власник (або ж уповноважена ним особа) явно або самоочевидно накладає набір певних правил по роботі з ним. Навмисне їх порушення і класифікується як *атака на інформацію* [34–37].

З 1991 р. за класифікатором Інтерполу навмисні дії, спрямовані на розкрадання або руйнування інформації в інформаційних системах і мережах, вчинені з корисливих чи хуліганських міркувань (*інформаційні злочини*) поділяються на наступні підвиди: несанкціонований доступ та перехоплення даних (*QA*); модифікація комп'ютерних даних (QD); комп'ютерне шахрайство (*QF*); незаконне копіювання (*QR*); комп'ютерний саботаж (*QS*); інші комп'ютерні злочини (*QZ*).

З масовою комп'ютеризацією усіх сфер діяльності людини обсяг інформації, що зберігається у цифровому вигляді, зріс у тисячі разів. Це, в свою чергу, значно підвищило ризик *витоку інформації*, у результаті чого остання стає відомою (доступною) суб'єктам, які не мали права доступу до неї. З розвитком комп'ютерних і стільникових мереж навіть відсутність фізичного доступу до деякої обчислювальної системи перестала бути гарантією збереженості інформації.

Розглянемо можливі наслідки атак на інформацію. У першу чергу, звичайно ж, це економічні втрати:

- розкриття комерційної інформації може призвести до значних прямих збитків;
- відомості про витік навіть незначного обсягу інформації суттєво впливає на репутацію організації, опосередковано призводячи до втрат в обсягах торгових операцій;
- якщо викрадення даних залишилося непоміченим, конкуренти можуть скористатися цим з метою повного розорення організації, нав'язуючи тій фіктивні або збиткові угоди;
- до збитків може призвести і проста підміна інформації як на етапі її передавання, так і на етапі зберігання всередині організації.

Цілком очевидно, що атаки на конфіденційну інформацію можуть заподіяти і величезну моральну шкоду її власникам.

В Україні в повному обсязі присутні всі ключові «класичні» кіберзлочини (шахрайство, здирництво, несанкціонований доступ до персональної інформації користувачів та автоматизованих баз даних, поширення порнографії, продаж зброї чи наркотиків тощо) і щороку їх кількість лише зростає [113]. Розглядаючи динаміку кількості карних справ, порушених Службою безпеки України за фактами виявлених кіберзлочинів, чітко прослідковується їх істотне збільшення: від 39 справ у 2005 до 158 у 2011 році. За результатами розгляду кримінальних справ у судових засіданнях за вказаний час винесено 20

судових вироків, з них у другому півріччі 2012 року — 14, у першому півріччі 2013 року — 6. Ще наочнішими є дані МВС України. Стрімко зростає кількість шахрайств, здійснюваних з використанням інформаційних технологій: лише за 6 місяців 2013 року їх було виявлено 986, у той час як за весь 2012 рік — 1663. Зростає кількість злочинів, пов'язаних з незаконними діями з документами на переказ, платіжними картками та іншими засобами доступу до банківських рахунків, обладнанням для їх виготовлення: за весь 2012 р. було виявлено 21 і стільки ж — вже за першу половину 2013 року.

У структурі злочинів переважають різноманітні випадки шахрайств, основною жертвою яких є банківсько-фінансовий сектор та його клієнти. Так, у 2013 р. кіберзлочинці намагалися привласнити з банківських рахунків 87 млн. ₴., «вдалими» стали шахрайства на 10 млн. ₴. Хоча в Україні й спостерігаються доволі високі показники розкриття кіберзлочинів у банківській сфері — до 80% вкрадених коштів повертається господарям, — при оцінюванні рівня розкриваності фінансових кіберзлочинів варто зважати на високий рівень їх латентності — банківським установам переважно вигідніше приховати факт зламу інформаційної системи, ніж заявляти про це у правоохоронні органи.

Зростають масштаби як традиційного маніпуляцій з платіжними картками, так і складніших кіберзлочинів. Крім того, й досі значними за обсягами та збитками залишаються такі злочини, як поширення порнографії, порушення авторських прав тощо.

Таким чином можна констатувати, що далеко не завжди держава реально обізнана з масштабами кіберзлочинності. І ця проблема наявна не лише в Україні, а й у всіх державах, де кіберзлочинність набирає обертів. Зокрема, у тій же Німеччині рівень інформаційних злочинів теж знаходиться на підйомі [114]. У 2016 р. кількість скоєних кримінальних діянь з використанням інтернет-технологій сягнула майже 83 тис. випадків, тоді як роком раніше поліція зареєструвала «лише» 46 тис. кіберзлочинів. Окремо наголошується, що в 2016 р. на кількість злочинів у сфері комп'ютерної інформації (комп'ютерний саботаж, *DDoS*-атаки хакерів) у ФРН збільшилася на 25% (зареєстровано 4422 таких злочини), а рівень розкриття таких правопорушень знизився на 4.6%, опустившись до 22.1%.

За даними дослідницької групи *CNews Analytics*, що спеціалізується на дослідженнях ринків інформаційних технологій і телекомунікацій, глобальний економічний збиток від комп'ютерних злочинів, що так або інакше пов'язані з атакою на інформацію, у 2002 р. склав близько 49,2 млрд. $, у 2003 р. — 116,4 млрд. $, у 2004 р. — від 300 до 400 млрд. $ [38].

Дослідницький центр *DataPro Research Corp.* [39] наводить таку картину розподілу основних причин втрати чи пошкодження інформації в комп'ютерних мережах: 52% і 13% — ненавмисні і навмисні дії користувачів відповідно, 35% — різноманітні техногенні фактори. Що стосується умисних дій, то за даними того ж дослідження, головний мотив комп'ютерних злочинів — викрадення грошей з електронних рахунків (44%), далі йдуть викрадення секретної інформації (18%), пошкодження програмного забезпечення (16%), фальсифікація інформації (12%), замовлення послуг за чужий рахунок (10%). Серед тих, хто був виконавцем зазначених дій, — поточний кадровий склад установ (81%), сторонні особи (13%), колишні працівники цих самих установ (6%).

Зрозуміло, що в такій ситуації особлива увага повинна приділятися створенню інформаційних систем, захищених від різноманітних загроз. Але при цьому в ході проектування і створення таких систем виникає ціла низка проблем, основними з яких є: складність інтеграції певних функцій безпеки в елементи архітектури системи; складність формування вичерпного набору необхідних і достатніх вимог безпеки; відсутність загальноприйнятих методів проектування систем безпеки.

Особливостями проектування архітектури системи забезпечення безпеки інформації, як, вочевидь, й інших складних багатофакторних систем, є множинність параметрів, що мають бути при цьому враховані і які важко зафіксувати, постійне зростання кількості загроз інформаційній безпеці і т. п. При цьому більшість існуючих рішень, наприклад таких як екранування і *VPN*, спроможні усунути лише частину загроз забезпечення безпеки інформації, а використовувані методи в основному зводяться до застосування визначеного переліку продуктів.

1.2. Категорії інформаційної безпеки з позицій захисту автоматизованих систем від несанкціонованого доступу

Інформація з точки зору інформаційної безпеки характеризується наступними категоріями:

- *конфіденційність* — гарантія того, що конкретна інформація є доступною лише тому колу осіб, для якого вона є призначеною; порушення цієї категорії є *викраденням* або *розкриттям* інформації;
- *цілісність* — гарантія того, що на даний момент інформація існує в її початковому вигляді, тобто при її зберіганні чи передаванні не

було зроблено несанкціонованих змін; порушення даної категорії зветься *фальсифікацією повідомлення*;

- *автентичність* — гарантія того, що джерелом інформації є саме той суб'єкт, який заявлений її автором; порушення цієї категорії зветься *фальсифікацією автора повідомлення*;
- *апельованість* — гарантія того, що за потреби можна бути довести, що автором повідомлення є саме заявлений суб'єкт і ніхто інший; відмінність даної категорії від попередньої полягає в тому, що при підміні автора, хтось інший намагається заявити, що він є автором повідомлення, а при порушенні апельованості — сам автор намагається уникнути відповідальності за видане ним повідомлення.

Стосовно інформаційно-телекомунікаційних систем використовуються інші категорії:

- *надійність* — гарантія того, що система вестиме себе заплановано як в нормальному, так і у позаштатному режимах;
- *точність* — гарантія точного і повного виконання всіх команд;
- *контроль доступу* — гарантія того, що різні групи осіб мають різний доступ до інформаційних об'єктів, і ці обмеження доступу виконуються постійно;
- *контрольованість* — гарантія того, що у будь-який момент може бути виконана повноцінна перевірка довільного компонента програмного комплексу;
- *контроль ідентифікації* — гарантія того, що клієнт (адресат) є саме тим, за кого себе видає;
- *стійкість до навмисних збоїв* — гарантія того, що при навмисному внесенні помилок в межах заздалегідь обговорених норм система вестиме себе прогнозовано.

1.3. Варіанти захисту інформації в автоматизованих системах

Серед можливих методів (заходів) захисту конфіденційної інформації найпоширенішим на сьогодні є метод *криптографічного захисту*, під яким розуміється приховання сутності повідомлення за рахунок його шифрування (кодування) за певним алгоритмом, що має на меті зробити повідомлення незрозумілим для непосвячених в цей алгоритм і/або у зміст ключа, що має бути використаний при дешифруванні. Тим не менш, зазначений метод захисту є неефективним щонайменше з двох причин.

По-перше, зашифрована за допомогою більш-менш стійкої криптографічної системи інформація є недоступною (протягом часу, що визначається стійкістю криптосистеми) для ознайомлення без знання алгоритму і/або ключа. Наслідком цього стало те, що силові структури деяких країн застосовують адміністративні санкції проти так званої «стійкої криптографії», обмежуючи використання криптографічних засобів приватними і юридичними особами без відповідної на те ліцензії.

По-друге, слід звернути увагу на те, що криптографічний захист захищає лише вміст конфіденційної інформації. При цьому сама лише наявність шифрованої інформації здатна привернути увагу потенційного зловмисника. Останній, отримавши доступ до криптографічно захищеного блока інформації, за певної мотивації здатен перевести сумарну міць підконтрольних йому обчислювальних ресурсів на дешифрування захищених даних, і проблема інформаційної безпеки повертається до стійкості криптографічного коду.

На противагу вищезазначеному, *стеганографічний захист* забезпечує приховання самого факту існування конфіденційних відомостей при їх передаванні, зберіганні чи обробці. Під приховання факту існування розуміється не тільки унеможливлення виявлення в перехопленому повідомленні наявності іншого (вбудованого) повідомлення, але й взагалі зробити неможливим викликання на цей рахунок будь-яких підозр. Задача неможливості неавторизованого видобування інформації при цьому відступає на другий план і у більшості випадків розв'язується додатковим застосуванням стандартних криптографічних методів. Загальною рисою стеганографічних методів є те, що повідомлення, що підлягає прихованню, вбудовується до структури деякого не приваблюючого увагу об'єкта (так званого *контейнера*), який згодом транспортується (надсилається) адресатові відкритими каналами зв'язку.

Розділ 2. ОСОБЛИВОСТІ ПОБУДОВИ СТЕГАНОГРАФІЧНИХ СИСТЕМ

2.1. Предмет, термінологія і напрямки застосування комп'ютерної стеганографії

У ході дослідження стеганографії, стає очевидним, що вона, по суті, не є чимось новим. Задачі захисту інформації від неавторизованого доступу тим або іншим способом вирішувалися протягом всієї історії людства [2].

В останні два десятиріччя, завдяки масовому поширенню мультимедійних технологій і засобів телекомунікацій розвиток стеганографії вийшов на принципово новий етап, який фахівці називають *комп'ютерною стеганографією* (КС). Серед основних напрямків використання КС — приховання (шляхом вбудовування) повідомлень у цифрових даних, які, як правило, мають аналогову природу свого походження (мова, зображення, аудіо- чи відеозаписи). Також в якості контейнерів (або так званих «носіїв») можливе використання текстових файлів або виконувальних файлів програм [1, 3–5, 19, 20]. У найпростішому випадку, наприклад, бітами конфіденційного повідомлення можуть бути замінені найменш значущі біти структурних елементів (пікселів, часових відліків, коефіцієнтів частотного розкладу тощо тощо) певного мультимедійного контейнера, без жодної для останнього шкоди. При цьому, за звичних обставин, жоден сторонній спостерігач буде неспроможним виявити ані якісь втрати в якості контейнера, ані, власне, приховане повідомлення [3, 9, 14, 21–23, 27].

Таким чином, скажімо, медіа-дані, викладені на масових ресурсах або надіслані через мережі загального користування, потенційно можуть таємно містити важливу для певних кіл інформацію і при цьому не викликати жодних підозр у широкого загалу. Яскравим прикладом подібної ситуації може слугувати той факт, що після терактів 11 вересня 2001 року у США деякі світові ЗМІ навіть вказували на зазначену техніку приховання, як можливий засіб зв'язку між членами терористичних угруповань, що в такий спосіб координували між собою атаки на знак протесту проти впливу Заходу на світовий устрій [28–30].

Незважаючи на численні відкриті публікації та щорічні конференції, тривалий час стеганографія не мала усталеної термінології. З середини 80-х років минулого століття для опису моделі стеганографічної системи (скорочено — *стегосистеми* або, що на думку

авторів цієї книги є більш правильним, *стеганосистеми*, оскільки префікс «стего» у перекладі з латини означає «дах» або «черепиця» і дещо спотворює сутність даного поняття) використовувалася так звана *«проблема ув'язнених» (Prisoners Problem)*, запропонована у 1983 р. Г. Сіммонсом *(G. J. Simmons)* [7]. Основні ж поняття стеганографії були узгоджені тільки у 1996 р. на 1-й Міжнародній конференції з приховання даних — *Information Workshop on Information Hiding'96* [10]. Тим не менш, навіть таке першоутворююче поняття як *«стеганографія»* різними фахівцями трактується неоднаково. Наприклад, деякі з них під стеганографією розуміють лише приховане передавання інформації. Інші ж відносять до неї ще й такі додатки як, наприклад, метеорний радіозв'язок, радіозв'язок з псевдовипадковим перестроюванням частоти, широкосмуговий радіозв'язок тощо [31, 32]. А ось у роботі [5] приводиться наступне визначення *цифрової стеганографії:* «...наука про непомітне і надійне приховання одних бітових послідовностей в інших, що мають аналогову природу». Згадуванням про аналогову природу цифрових даних підкреслюється той факт, що інформація прихованого повідомлення вбудовується до структури оцифрованих безперервних сигналів. Таким чином, у порівнянні з ЦС комп'ютерна стеганографія має більш широкоохоплюючу сутність, оскільки нею розглядаються ще й такі питання, як введення даних до заголовків *IP*-пакетів, різноманітних текстових документів та файлів інших форматів.

Словом «непомітне» у наведеному вище визначенні ЦС підкреслюється обов'язкове включення до складу системи стеганографічного передавання даних людини, як індивідуального кінцевого одержувача. Тобто людина розглядається як специфічний приймач даних, який висуває до системи передавання достатньо важко формалізовані вимоги [5].

Таким чином, *стеганографічна система* або, скорочено, *стеганосистема* — це сукупність засобів і методів, які використовуються з метою формування прихованого (непомітного) каналу передавання інформації [10, 22]. Іншими словами, це система, що виконує задачу вбудовування і виокремлення одних інформаційних послідовностей з інших. Причому процес вбудовування даних, подібно до процесу стиснення (ущільнення), є відмінним від операції шифрування. Його метою є, скоріше, не обмеження чи регламентування доступу до сигналу (файлу) контейнера, а в значній мірі гарантування, що вбудовані дані будуть непомітними, залишаться непошкодженими (немодифікованими) і такими, що підлягають відновленню за наявної на це потреби [14].

При побудові довільної стеганосистеми зазвичай мають бути враховані наступні вимоги [3, 5, 11]:

- стеганосистема повинна мати прийнятну обчислювальну складність своєї реалізації (тут під обчислювальною складністю розуміється кількість кроків або арифметико-логічних операцій, достатніх для розв'язання поставленої перед системою задачі, у даному випадку — процесу вбудовування/видобування прихованої інформації до/зі структури контейнера);
- повинна забезпечуватися необхідна пропускна здатність (що є особливо актуальним для стеганосистем прихованого передавання даних);
- методи приховання мають забезпечувати автентичність і цілісність секретної інформації для авторизованої особи;
- потенційний порушник має повне уявлення про стеганосистему і деталі її реалізації. Єдине, що йому невідоме, — це ключ, за допомогою якого тільки його власник може встановити факт наявності та зміст прихованого повідомлення;
- якщо факт існування прихованого повідомлення стає відомим порушнику, це не повинне дозволити останньому видобути його доти, доки сам ключ зберігається в таємниці;
- порушник повинен бути позбавлений будь-яких технічних та інших переваг у детектуванні наявності і, тим паче, розкритті змісту секретних повідомлень.

На думку авторів [5], стеганографія включає в себе наступні напрямки:

- вбудовування інформації з метою її прихованого передавання;
- вбудовування цифрових водяних знаків (ЦВЗ);
- вбудовування ідентифікаційних номерів;
- вбудовування заголовків.

Аналіз інших літературних джерел, зокрема [3, 14, 16, 19–21], а також *Internet*-ресурсів дозволяє зробити висновок, що на сьогодні стеганосистеми активно застосовуються для розв'язання таких ключових завдань:

- захист конфіденційної інформації від несанкціонованого доступу;
- захист авторського права на інтелектуальну власність;
- подолання систем моніторингу і управління мережними ресурсами;
- «камуфлювання» програмного забезпечення;
- створення прихованих від законного користувача каналів витоку інформації.

Сфера *захисту конфіденційної інформації від несанкціонованого доступу* є найбільш ефективною при вирішенні проблеми захисту

секретної інформації. Наприклад, лише одна секунда стереозвуку, оцифрованого з частотою дискретизації 44 кГц та однобайтовим квантуванням відліків рівнів, за рахунок заміни найменш значущих молодших розрядів кодових комбінацій, що характеризують ці відліки, на біти прихованого повідомлення дозволяє приховати $2 \cdot 44/8 = 11$ кБ інформації (при загальному об'ємі аудіофайлу ~ 88 кБ). При цьому зміна результуючого рівня аудіосигналу, що відповідає модифікованому відліку, у найгіршому випадку становитиме $1/2^8 \approx 0.4\%$, що під час прослуховування не виявляється переважною більшістю людей. Якщо ж звук аудіоконтейнера буде 16-бітним, то зміна рівнів взагалі стає меншою за $1/2^{16} \approx 0{,}002\%$.

Іншим, не менш важливим завданням стеганографії є *захист авторських прав* (*прав інтелектуальної власності*) на той чи інший мультимедійний продукт від так званого «піратства». При цьому на дані, що підлягають захисту, наноситься спеціальна мітка, яка залишається невидимою або невідчутною для людини, але розпізнається спеціалізованим програмним забезпеченням.

Стеганографічні методи, спрямовані на *протидію системам моніторингу й управління мережними ресурсами промислового шпигунства*, дозволяють протистояти спробам контролю над інформаційним простором при проходженні інформації через сервери управління локальних і глобальних обчислювальних мереж.

Ще однією областю використання стеганосистем є *камуфлювання програмного забезпечення*. У тих випадках, коли використання останнього незареєстрованними користувачами є небажаним, воно може бути закамуфльованим під стандартний універсальний програмний продукт (наприклад, текстовий редактор абощо) чи прихованим у мультимедійних файлах (наприклад, у звуковому супроводі до відео).

Наскільки б не відрізнялися між собою напрямки використання стеганографії, вимоги, що при цьому до них висуваються, багато у чому залишаються незмінними (див. вище). Але існують й відхилення від загальноприйнятих правил. Так, наприклад, відмінність постановки задачі прихованого передавання даних від постановки задачі вбудовування ЦВЗ полягає у тому, що у першому випадку для встановлення факту наявності і видобування прихованого повідомлення порушник має докласти певних зусиль, тоді як у другому випадку існування такого повідомлення (ЦВЗ) не приховується. Більше того, порушник на законних підставах може мати пристрій його виявлення (наприклад, у складі *BD*-програвача).

Потенційно можливі сфери використання стеганографії представлено на рис. 2.1.

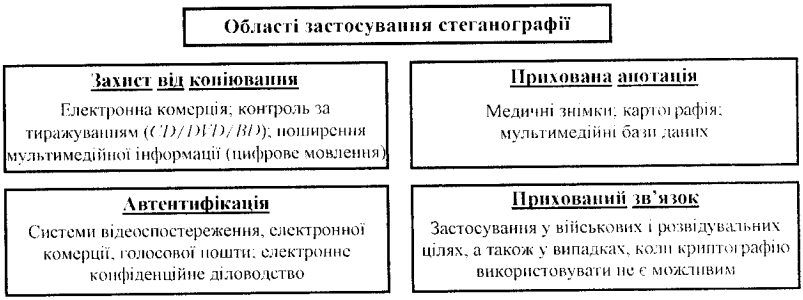


Рис. 2.1. Потенційні області застосування комп'ютерної стеганографії

2.2. Проблема стійкості стеганографічних систем

Кожне з зазначених на рис. 2.1 завдань вимагає певного співвідношення між розміром (об'ємом) вбудованого повідомлення і його стійкістю до зовнішніх впливів. Для більшості сучасних методів КС, що використовуються для приховання повідомлень у різноманітних мультимедійних файлах, має місце представлена на рис. 2.2 залежність надійності стеганосистеми від об'єму вбудовуваних за її допомогою даних, з якої видно, що при збільшенні останнього надійність системи суттєво погіршується.

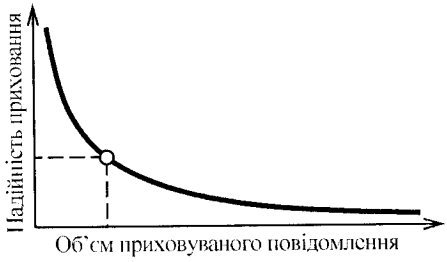


Рис. 2.2. Взаємозв'язок між стійкістю стеганосистеми та об'ємом прихованого повідомлення при незмінній ємності контейнера

Таким чином, існує перспектива прийняття оптимального рішення при виборі між кількістю (об'ємом) прихованих даних і ступенем стійкості (прихованості) до можливої модифікації чи аналізу контейнера. Шляхом обмеження ступеня погіршення якостей контейнера, що їх здатна сприймати людина, при стеганографічній його обробці

можна досягти або високого рівня (об'єму) вбудованих даних, або високої стійкості до модифікацій (аналізу), але в жодному разі не обох цих показників одночасно, оскільки зростання одного з них неминуче призводить до зменшення іншого. Не зважаючи на те, що дане твердження математично може бути продемонстроване лише для деяких методів стеганографії (наприклад, для приховання шляхом розширення спектра), очевидно, що воно є справедливим і для інших методів КС [14]. У будь-якому випадку, завдяки надлишковому характеру інформації, що становить собою контейнер, є можливість підвищити ступінь надійності приховання, жертвуючи при цьому пропускною здатністю (об'ємом прихованих даних). Об'єм вбудованих даних і ступінь модифікації контейнера змінюються від методу до методу. Також є очевидним і той факт, що в залежності від цілей, для яких використовується стеганографічне приховання даних, різними будуть і вимоги стосовно рівня стійкості системи до модифікації контейнера. Як наслідок — для різних вирішуваних задач є оптимальними й різні методи стеганографії.

Процес проведення стеганоаналізу полягає в оцінці перехопленого контейнера на предмет наявності в ньому прихованого повідомлення. Наявність же всередині цифрового носія вбудованої інформації зумовлює зміни (перебудови) властивостей контейнера, що в тій або іншій мірі призводить до погіршення його характеристик або до набуття цими характеристиками невластивих їм значень. Ці характеристики можуть виступати свого роду «маркерами», що сигналізують про здійснення певних маніпуляцій по заповненню контейнера повідомленням і, таким чином, основна ідея стеганографії — приховання факту існування секретної інформації — не дотримується.

Стеганоаналіз на предмет заповненості контейнера прихованою інформацією може набувати різних форм: виявлення наявності (детектування), видобування і, зрештою, видалення чи руйнування прихованих даних [40]. Крім того, порушник може замість вже існуючої прихованої інформації вбудувати певну дезінформацію. Детальніше атаки на стеганосистеми і протидія їм розглядатимуться у розділі 3.

2.3. Структурна схема і математична модель типової стеганосистеми

У загальному випадку стеганографічна система може бути представлена як система зв'язку [5, 115, 116]. Узагальнена структурна схема типової стеганосистеми зображена на рис. 2.3.

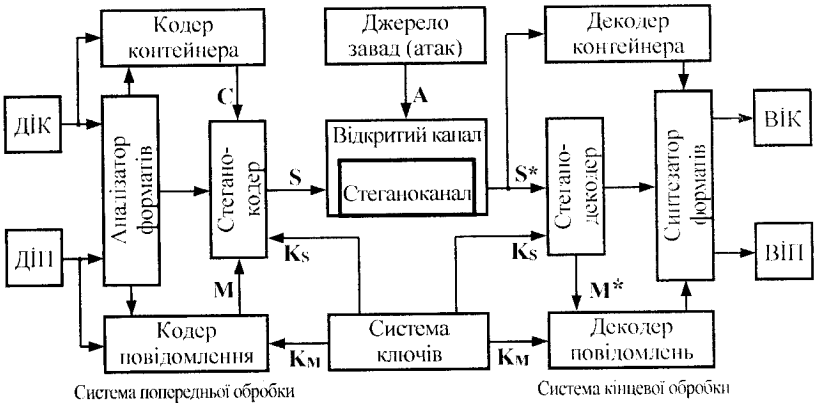


Рис. 2.3. Структурна схема комп'ютерної стеганосистеми як системи зв'язку

Основними стеганографічними поняттями є повідомлення і контейнер. *Стеганографічним повідомленням (стеганоповідомленням)* $\boldsymbol{m} \in \mathbf{M}$ є секретна інформація, наявність якої необхідно приховати, $\mathbf{M} = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n\}$ — множина всіх повідомлень. *Стеганографічним контейнером (стеганоконтейнером)* $\boldsymbol{c} \in \mathbf{C}$ зветься несекретна (відкрита широкому загалу) інформація, яку можна використати для приховання повідомлення шляхом маскування, $\mathbf{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_q\}$ — множина всіх контейнерів, причому між потужностями множин **C** та **M** має бути дотримане наступне співвідношення: $\boldsymbol{q} >> \boldsymbol{n}$. Тобто приховання інформації, яка здебільшого має досить великий об'єм, висуває суттєві вимоги до контейнера, розмір якого повинен щонайменше в декілька разів перевищувати розмір вбудовуваних даних. Зрозуміло, що для підвищення прихованості зазначене співвідношення повинне бути якомога більшим.

Узагалі, в якості повідомлення і контейнера можуть виступати мультимедійні дані довільного формату і, до того ж, у будь-якому поєднанні. Контейнер, що не містить у собі прихованої інформації, називається *пустим* (так званий *контейнер-оригінал*), а контейнер, у якому вже приховано повідомлення (*стеганограма*), — *заповненим* (*контейнером-результатом*). Одна з вимог, що при цьому обов'язково має виконуватися: контейнер-результат не повинен бути відмінним від контейнера-оригіналу (візуально, на слух тощо). Файл (об'єкт), отриманий за результатами вбудовування до себе повідомлення, зветься *стеганофайлом* (стеганооб'єктом). Наприклад, при вбудовуванні повідомлення у файл зображення одержують *стеганозображення*,

вбудовування до текстового файлу має підсумком отримання *стеганотексту* і т. п.

Виокремлюють два основних типи контейнерів: потоковий і фіксований. *Потоковий контейнер* являє собою послідовність бітів, що безперервно змінюється і не поділяється на блоки або якісь інші подібні структури фіксованої довжини. До такого контейнера повідомлення пакується у реальному масштабі часу, так що в кодері невідомо заздалегідь: чи вистачить розмірів контейнера для передавання всього повідомлення. До одного контейнера великого розміру може бути вбудовано декілька повідомлень. Інтервали між заповнюваними структурними елементами контейнера можуть визначатися генератором псевдовипадкової послідовності (ПВП), що працює за заданим законом розподілу. Основна проблема полягає у здійсненні синхронізації, визначенні початку і кінця послідовності. Якщо в даних контейнера існують біти синхронізації, заголовки пакетів і т. п., то прихована інформація може йти відразу після них. Важкість забезпечення синхронізації є перевагою з точки зору забезпечення прихованості передавання. Нажаль, на сьогодні практично відсутні роботи, присвячені розробці стеганосистем з потоковим контейнером. Як приклад перспективної реалізації такого типу контейнера можна навести мобільний додаток до телефону, який під прикриттям пересічної, незначущої телефонної розмови дозволятиме передавати іншу, конфіденційну розмову, дані тощо. Не знаючи секретного ключа важко не тільки довідатися про зміст прихованого передавання, але й про сам факт його здійснення.

У *фіксованого контейнера* розміри і характеристики є заздалегідь відомими (принаймні, на передавальному боці). Це дозволяє здійснювати вкладення даних більш оптимальним (у певному сенсі) чином. Надалі у книзі розглядатимуться переважно фіксовані контейнери (у подальшому — просто «контейнери»).

Сам контейнер може бути обраним, випадковим або нав'язаним. *Обраний* контейнер залежить від вбудовуваного повідомлення, а в граничному випадку є його функцією. Такий тип контейнера більше характерний саме для стеганографії. *Нав'язаним* контейнер стає у тому випадку, коли особа, що його надає, підозрює про можливий стеганографічний обмін інформацією і бажає йому зашкодити або використати у власних цілях. На практиці частіше за все мають справу з *випадковими* (*ситуативними*) контейнерами [5].

На рис. 2.3 сигнал з виходу джерела інформації-контейнера (ДІК), який повинен бути відкритим чином переданий системою зв'язку і попередньо може мати певний формат, надходить на входи аналізатора форматів (АФ) і кодера контейнера у системі попередньої обробки

(СПО). Крім цього на входи АФ і кодера повідомлення СПО подається сигнал з виходу джерела інформації-повідомлення (ДІП), який також може мати довільний формат. Встановивши структуру інформації на своїх входах, АФ видає керувальні сигнали на входи відповідних джерелам кодерів стосовно наявності або відсутності потреби у застосуванні операцій додаткового кодування (перекодування). При цьому до задач кодерів контейнера і повідомлення входить забезпечення сумісності форматів даних ДІК та ДІП із засобами подальшої цифрової обробки у стеганокодері, за допомогою якого здійснюється вбудовування повідомлення **M** до контейнера **C**.

Кодер повідомлення крім узгодження форматів може виконувати й інші корисні функції. Перед вбудовуванням особливо важливого повідомлення до контейнера, попередньо його можна зашифрувати стійким криптографічним кодом — при цьому повинні бути заздалегідь згенеровані, зареєстровані і поширені між авторизованими учасниками інформаційного обміну ключі шифрування/дешифрування повідомлення $\mathbf{K}_M$. Крім того, при організації процесу вбудовування/видобування повідомлення до/з контейнера може знадобитися система стеганоключів $\mathbf{K}_S$. Алгоритми і протоколи організації умовного допуску можуть відповідати процедурам, наведеним, зокрема, у [65, 117, 118] або ж регламентованим стандартами тієї системи зв'язку, на базі якої створюється стеганосистема. За аналогією з криптографією, тип ключа обумовлює існування двох типів стеганосистем:

- *з секретним ключем* (використовується один ключ, який визначається до початку обміну стеганограмою або передається захищеним каналом);
- *з відкритим ключем* (для пакування і розпакування повідомлення застосовуються різні ключі, які різняться таким чином, що за допомогою обчислень неможливо вивести один ключ з іншого і тому один з ключів (відкритий) може вільно передаватися незахищеним каналом).

В якості секретного алгоритму може бути використаний генератор ПВП. Якісний генератор ПВП, орієнтований на використання в системах захисту інформації повинен задовольняти певним вимогам [42], серед яких:

- *криптографічна стійкість* (відсутність у порушника можливості передбачити наступний біт на основі відомих йому попередніх з імовірністю, що відрізняється від ½. На практиці криптографічна стійкість оцінюється статистичними методами. Зокрема, Національним Інститутом Стандартів і Технологій США (НІСТ) розроблено Керівництво з проведення статистичних випробувань

генераторів ПВП, орієнтованих на використання в задачах криптографічного захисту інформації [43]);

- *гарні статистичні властивості*, за якими ПВП не повинна суттєво відрізнятися від істинно випадкової послідовності;
- великий *період формованої послідовності*;
- ефективна *апаратна/програмна реалізація*.

Крім того, статистично (криптографічно) безпечний генератор ПВП має задовольняти наступним вимогам:

- жоден статистичний тест не спроможний виявити у ПВП аніяких закономірностей, іншими словами він не відрізняє цю послідовність від істинно випадкової;
- при ініціалізації випадковими значеннями, генератор породжує статистично незалежні псевдовипадкові послідовності.

В якості основи генератора ПВП може використовуватися, наприклад, лінійний рекурентний регістр. Тоді учасникам прихованого інформаційного обміну необхідно заздалегідь домовитися про початкове заповнення цього регістра. Числа, породжувані генератором ПВП, можуть визначати позиції модифікованих відліків у випадку фіксованого контейнера або інтервали між ними у випадку потокового контейнера. Слід зауважити, що метод випадкового обрання величини інтервалу між заповнюваними елементами контейнера не є достатньо ефективним з двох причин. По-перше, приховані дані мають бути розподілені по всьому контейнеру. Тому рівномірний розподіл довжин інтервалів (від найменшого до найбільшого) може бути досягнутий лише наближено, оскільки повинна існувати впевненість у тому, що все повідомлення зможе бути вбудоване, тобто вміститься у контейнер. По-друге, довжини інтервалів між відліками шуму (у багатьох моделях стеганосистем сигнал-контейнер розглядається саме як адитивний шум [44]) розподілені не за рівномірним, а за експонентним законом. Генератор ПВП з експонентно розподіленими інтервалами є складним в реалізації.

Прихована інформація заноситься у відповідності з ключем до тих бітів, модифікація яких не призводить до істотних спотворень контейнера. Ці біти утворюють так званий *стеганошлях*. При цьому під «істотним» розуміється таке спотворення, що веде до зростання імовірності виявлення факту наявності прихованого повідомлення після проведення стеганоаналізу. Кількість заміщених у стеганоконтейнері бітів, виражена у відсотках відносно загального об'єму останнього, зветься *інформаційною ємністю стеганошляху*.

У стеганокодері, згідно з використовуваним стеганографічним алгоритмом (СА) і, можливо, системою ключів **Ks**, повідомлення **M**

заноситься до тих елементів контейнера **C**, модифікація яких відповідає цілям, поставленим перед стеганосистемою (див. рис. 2.1). Модифіковані (заповнені) елементи контейнера утворюють *стеганографічний канал* (*стеганоканал*), створений на основі відкритого каналу, на вхід якого подається заповнений контейнер **S**. Саме алгоритм (стратегія) внесення елементів повідомлення до контейнера визначає методи стеганографії, які в свою чергу поділяються на певні групи, наприклад, в залежності від того, файл якого формату було обрано в якості контейнера. Взагалі ж, існування стеганоканалу, як, власне кажучи, і одержувача стеганограми є найбільш узагальнюючим випадком, оскільки заповнений контейнер може, наприклад, зберігатися у «відправника», який просто поставив собі за мету обмежити неавторизований доступ до певної інформації. У такому випадку відправник виступає і в ролі одержувача.

Під час перебування у стеганоканалі контейнер, що містить приховане повідомлення, може піддаватися навмисним атакам або випадковим завадам. Тому в деяких випадках може виявитися необхідною підвищена стійкість стеганограми до спотворювань **A**, що вносяться джерелом завад і атак. Потік даних, використовуваний в якості контейнера, вже під час передавання може зазнавати різноманітних трансформацій: стиснення (у тому числі з використанням алгоритмів із втратою даних), перетворення на інший формат тощо. Тому для збереження цілісності вбудованого повідомлення можна застосувати контроль кодом *CRC* або будь-який інший відомий спосіб завадостійкого кодування, наприклад з представлених у [65, 115].

У стеганодетекторі здійснюється визначення наявності в контейнері (можливо вже спотвореному) прихованих даних. Спотворення може бути обумовлене впливом помилок в каналі зв'язку, операцій обробки сигналу, навмисних атак порушників тощо. Як вже зазначалося, у багатьох моделях стеганосистем сигнал-контейнер розглядається як адитивний шум. Тоді задача виявлення і виділення стеганограми є класичною для теорії зв'язку [115]. Але такий підхід не враховує двох факторів: невипадкового характеру контейнера і вимог по збереженню його якості. Ці моменти не зустрічаються у відомій теорії виявлення і виділення сигналів на фоні адитивного шуму. Очевидно, що їх врахування дозволить побудувати більш ефективні стеганосистеми.

Розрізнюють стеганодетектори, призначені лише для виявлення факту наявності вбудованого повідомлення, і пристрої, призначені для виділення цього повідомлення з контейнера — *стеганодекодери*.

У стеганодекодері здійснюється видобування з контейнера **S*** (можливо, зміненому під час проходження відкритим каналом) даних

стеганограми **M***. У стеганосистемах зі сфери захисту від копіювання (рис. 2.1), метою яких є забезпечення можливості підтвердження факту наявності вбудованого повідомлення, цей структурний блок може виконувати функції детектора ЦВЗ (стеганодетектора). Декодовані контейнер і повідомлення надходять на синтезатор форматів відтворення (СФВ) у складі системи кінцевої обробки (СКО). Призначенням СФВ є остаточне підготування вказаних елементів (зокрема, регулювання динамічного діапазону, форматування або конвертація повідомлень тощо) до їх відображення відповідними відтворювачами інформації (ВІК та ВІП) кінцевих приймачів системи зв'язку.

Отже, в стеганосистемі відбувається об'єднання двох типів інформації таким чином, щоб вони по-різному сприймалися принципово різними детекторами. В якості одного з детекторів виступає система виділення прихованого повідомлення, в якості іншого — людина. Алгоритм вбудовування повідомлення в найпростішому випадку складається з двох основних етапів:

1) вбудовування в стеганокодері секретного повідомлення до контейнера-оригіналу;

2) виявлення (виділення) в стеганодетекторі прихованого зашифрованого повідомлення з контейнера-результату.

Виходячи з цього, розглянемо *математичну модель стеганосистеми*. Процес тривіального *стеганографічного перетворення* описується наступними залежностями:

$$\boldsymbol{E}\colon\ \mathbf{C} \times \mathbf{M} \to \mathbf{S}; \tag{2.1}$$

$$\boldsymbol{D}\colon\ \mathbf{S} \to \mathbf{M}, \tag{2.2}$$

де $\mathbf{S} = \{(\boldsymbol{c}_1, \boldsymbol{m}_1), (\boldsymbol{c}_2, \boldsymbol{m}_2), \ldots, (\boldsymbol{c}_n, \boldsymbol{m}_n), \ldots, (\boldsymbol{c}_q, \boldsymbol{m}_q)\} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_q\}$ — це множина контейнерів-результатів.

Вираз (2.1) описує процес приховання інформації, вираз (2.2) — видобування прихованої інформації. Необхідною умовою при цьому є відсутність так званого «перетинання» [3]: тобто, якщо $\boldsymbol{m}_a \neq \boldsymbol{m}_b$, причому $\boldsymbol{m}_a, \boldsymbol{m}_b \in \mathbf{M}$, а $(\boldsymbol{c}_a, \boldsymbol{m}_a), (\boldsymbol{c}_b, \boldsymbol{m}_b) \in \mathbf{S}$, то $\boldsymbol{E}(\boldsymbol{c}_a, \boldsymbol{m}_a) \cap \boldsymbol{E}(\boldsymbol{c}_b, \boldsymbol{m}_b) = \varnothing$. Крім того, необхідно, щоб потужність множин задовольняла нерівності $|\mathbf{C}| \geq |\mathbf{M}|$. При цьому обидва адресати (відправник і одержувач) повинні знати алгоритм прямого ($\boldsymbol{E}$) та зворотного ($\boldsymbol{D}$) стеганографічного перетворення.

Отже, в загальному випадку *стеганосистема* — це сукупність $\Sigma = (\mathbf{C}, \mathbf{M}, \mathbf{S}, \boldsymbol{E}, \boldsymbol{D})$ контейнерів (оригіналів і результатів), повідомлень і перетворень, що їх пов'язують між собою.

Для більшості стеганосистем множина контейнерів **C** обирається таким чином, щоб в результаті стеганографічного перетворення (2.1) заповнений контейнер і контейнер-оригінал були подібними, що формально може бути оцінене за допомогою функції подібності [3].

Означення 2.1

Нехай **C** — непорожня множина, тоді функція $sim(\mathbf{C}) \to (-\infty, 1] \in$ *функцією подібності* на множині **C**, якщо для будь-яких $x, y \in \mathbf{C}$ справедливо, що $sim(x, y) = 1$ у випадку $x = y$ і $sim(x, y) < 1$ при $x \neq y$.

Стеганосистема може вважатися *надійною*, якщо для всіх $m \in \mathbf{M}$ і $c \in \mathbf{C}$ виконується умова $sim[c, E(c, m)] \approx 1$. Причому в якості контейнера c повинен обиратися такий, що раніше не використовувався. Крім того, неавторизована особа не повинна мати доступ до набору контейнерів, що використовуються для секретного зв'язку.

Обрання визначеного контейнера c з набору можливих контейнерів **C** може здійснюватися довільно (так званий *сурогатний метод* вибору контейнера) або ж шляхом обрання найбільш придатного, який менше за інших зміниться під час стеганоперетворення (*селективний метод*). В останньому випадку контейнер обирається відповідно до правила:

$$c = \max_{x \in C} sim\left[x, E(x, m)\right]. \qquad (2.3)$$

Також слід зазначити, що функції прямого (E) і зворотного (D) стеганоперетворень у загальному випадку можуть бути довільними (але, звичайно, відповідними одна одній), однак на практиці вимоги до стійкості прихованої інформації накладають на них певні обмеження. Так, у переважній більшості випадків $E(c, m) \approx E(c + \delta, m)$ або $D[E(c, m)] \approx D[E(c + \delta, m)] = m$. Тобто незначно модифікований контейнер (на величину δ) не призводить до зміни прихованої в ньому інформації [5].

2.4. Протоколи стеганографічних систем

Важливе значення для досягнення цілей стеганографії мають *протоколи*. Під протоколом розуміється «порядок дій, що вживаються двома чи більше сторонами, призначений для вирішення певної задачі» [11]. Можна розробити винятково ефективний алгоритм приховання інформації, але через його неправильне застосування так і не досягти поставленої мети.

І протокол, і алгоритм є певною послідовністю дій. Відмінність між ними полягає в тому, що до протоколу повинні бути обов'язково

залучені двоє або більше сторін. При цьому припускається, що учасники беруть на себе зобов'язання дотримуватись протоколу. Так само, як і алгоритм, протокол складається з кроків. На кожному кроці протоколу виконуються певні дії, які можуть полягати, наприклад, у проведенні деяких обчислень, або ж у здійсненні якихось дій.

Як уже зазначалося в попередньому підрозділі, в стеганографії розрізнюють системи з секретним і відкритим ключами. У перших використовується один ключ, що повинен бути заздалегідь відомий авторизованим абонентам до початку прихованого обміну секретними повідомленнями (або ж пересланий захищеним каналом під час зазначеного обміну). У системах з відкритим ключем для вбудовування і видобування прихованої інформації використовуються різні, не вивідні один з одного ключі — відкритий і секретний.

З огляду на велике різноманіття стеганографічних систем, доцільно звести їх до наступних чотирьох типів [3]:

- *безключові стеганосистеми*;
- *стеганосистеми з секретним ключем*;
- *стеганосистеми з відкритим ключем*;
- *змішані стеганосистеми*.

2.4.1. Безключові стеганосистеми

Для функціонування безключових стеганографічних систем крім алгоритму стеганографічного перетворення відсутня необхідність у жодних додаткових даних на зразок стеганоключа.

Означення 2.2

Сукупність $\Sigma = (\mathbf{C}, \mathbf{M}, \mathbf{S}, E, D)$, де $\mathbf{C}$ — множина контейнерів-оригіналів; $\mathbf{M}$ — множина секретних повідомлень, причому $|\mathbf{M}| \le |\mathbf{C}|$; $\mathbf{S}$ — множина контейнерів-результатів, причому $sim(\mathbf{C}, \mathbf{S}) \to 1$; E: $\mathbf{C} \times \mathbf{M} \to \mathbf{S}$ та D: $\mathbf{S} \to \mathbf{M}$ — відповідно функції прямого (вбудовування) і зворотного (видобування) стеганоперетворення, причому $D[E(c, m)] = m$ для будь-яких $m \in M$ і $c \in C$, зветься *безключовою стеганографічною системою*.

Таким чином, безпека безключових стеганосистем базується лише на таємності використовуваних стеганографічних перетворень E і D. Це суперечить визначальному принципові, встановленому О. Керкгоффсом для системи захисту інформації [6], оскільки стійкість останньої залежатиме лише від ступеню поінформованості порушника щодо функцій E і D.

Для підвищення безпеки безключових систем, перед початком процесу стеганографічного приховання попередньо виконується криптографічне шифрування прихованої інформації. Цілком очевидно, що такий підхід збільшує захищеність усього процесу зв'язку, оскільки ускладнює виявлення прихованого повідомлення. Однак «сильні» стеганосистеми, як правило, здатні виконувати покладені на них функції без попереднього криптографічного захисту вбудовуваного повідомлення.

2.4.2. Стеганосистеми з секретним ключем

За принципом Керкгоффса, безпека системи повинна ґрунтуватися на певному фрагменті секретній інформації — ключі, що (як правило, попередньо) розподіляється між авторизованими особами. Відправник, вбудовуючи секретне повідомлення до обраного контейнера c, використовує стеганоключ k. Якщо даний ключ є відомим одержувачеві, то він зможе видобути з контейнера приховане повідомлення. Без знання ключа будь-яка стороння особа цього зробити не зможе.

Означення 2.3

Стеганосистемою з секретним ключем називається сукупність $\Sigma = (\mathbf{C}, \mathbf{M}, \mathbf{K}, \mathbf{S}^{\mathbf{K}}, E, D)$, де $\mathbf{C}$ — множина контейнерів-оригіналів; $\mathbf{M}$ — множина секретних повідомлень, причому $|\mathbf{M}| \le |\mathbf{C}|$; $\mathbf{S}^{\mathbf{K}}$ — множина контейнерів-результатів, причому $sim(\mathbf{C}, \mathbf{S}^{\mathbf{K}}) \to 1$; $\mathbf{K}$ — множина секретних стеганоключів; E: $\mathbf{C} \times \mathbf{M} \times \mathbf{K} \to \mathbf{S}^{\mathbf{K}}$ та D: $\mathbf{S}^{\mathbf{K}} \times \mathbf{K} \to \mathbf{M}$ — функції прямого і зворотного стеганоперетворення з властивістю $D[E(c, m, k), k] = m$ для будь-яких $m \in \mathbf{M}$, $c \in \mathbf{C}$ і $k \in \mathbf{K}$.

Даний тип стеганографічних систем припускає наявність безпечного (захищеного) каналу обміну стеганоключами.

Іноді ключ k обчислюють за допомогою секретної хеш-функції (*hash function*), використовуючи певні характерні особливості контейнера. Якщо стеганоперетворення E у підсумку не змінює ці обрані особливості, то одержувач також зможе обчислити стеганоключ (хоча й у цьому випадку захист залежатиме від таємності хеш-функції, і, таким чином, знову порушується принцип Керкгоффса). Очевидно, що для досягнення адекватного рівня захисту, таку особливість у контейнері необхідно вибирати досить уважно.

У деяких СА під час видобування прихованої інформації додатково використовують відомості про первинний контейнер або деякі інші дані, які відсутні у стеганограмі. Такі системи становлять обмежений інтерес, оскільки вони вимагають передавання початкового вигляду контейнера, що еквівалентно традиційній задачі ключового обміну. Подібні алгоритми можуть бути відзначені як окремий випадок

стеганосистем з секретним ключем, у яких $\mathbf{K} = \mathbf{C}$ або $\mathbf{K} = \mathbf{C} \times \mathbf{K}'$, де $\mathbf{K}'$ — множина додаткового набору секретних ключів.

2.4.3. Стеганосистеми з відкритим ключем

Стеганографія з відкритим ключем спирається на досягнення криптографії останніх 40 років. Стеганографічні системи з відкритим ключем не мають потреби у додатковому каналі ключового обміну. Для їхнього функціонування необхідно мати два стеганоключі: один *секретний*, що його необхідно зберігати в таємниці, а інший — *відкритий*, що може зберігатися в доступному для всіх місці. При цьому відкритий ключ використовується для вбудовування повідомлення, а секретний — для його видобування.

Означення 2.4

Стеганосистемою з відкритим ключем називається сукупність $\Sigma = (\mathbf{C}, \mathbf{M}, \mathbf{K}, \mathbf{S}^{\mathbf{K}}, E, D)$, де $\mathbf{C}$ — множина контейнерів-оригіналів; $\mathbf{M}$ — множина секретних повідомлень, $|\mathbf{M}| \le |\mathbf{C}|$; $\mathbf{S}^{\mathbf{K}}$ — множина контейнерів-результатів, $sim(C, \mathbf{S}^{\mathbf{K}}) \to 1$; $\mathbf{K} = (k_в, k_с)$ — множина пар стеганоключів (відкритий ключ $k_в$ використовується для приховання інформації, а секретний ключ $k_с$ — для її видобування); E: $\mathbf{C} \times \mathbf{M} \times k_в \to \mathbf{S}^{\mathbf{K}}$ та D: $\mathbf{S}^{\mathbf{K}} \times k_с \to \mathbf{M}$ — функції прямого і зворотного стеганоперетворення з властивістю $D[E(c, m, k_в), k_с] = m$ для будь-яких $m \in \mathbf{M}$, $c \in \mathbf{C}$ і $k_в, k_с \in \mathbf{K}$.

Слід зауважити, що стеганоключ не шифрує дані, а приховує місце їх вбудовування у контейнері. Приховані дані можуть бути додатково зашифровані класичними методами, але безпосередньо стеганографії це питання вже не стосується.

Стеганосистеми з відкритими ключами використовують той факт, що функція видобування прихованих даних D може бути застосована до будь-якого контейнера незалежно від того, знаходиться в ньому приховане повідомлення чи ні (c_i або s_i). Якщо приховане повідомлення відсутнє, на виході одержуватиметься певна послідовність випадкових символів з використовуваного алфавіту. Якщо ця послідовність статистично не відрізнятиметься від шифртексту криптосистеми з відкритим ключем, тоді в безпечній стеганосистемі можна приховувати отриманий у такий спосіб шифртекст, а не відкритий [3].

2.4.4. Змішані стеганосистеми

На практиці перевага віддається безключовим стеганосистемам, хоча останні і можуть бути відразу розкриті у випадку, якщо поруш-

ник дізнається про метод стеганоперетворення, що був при цьому застосований. У зв'язку з цим у безключових системах часто використовують особливості криптографічних систем з відкритим і/або секретним ключем [3, 12].

З огляду на велику різноманітність форматів, які можуть мати приховані повідомлення і контейнери (текст, зображення, звук або відео, що у свою чергу також поділяються на відповідні підформати), доцільним бачиться попереднє перетворення прихованого повідомлення на зручний для вбудовування і оптимальний з точки зору рівня прихованості в заданому контейнері формат [5]: U: $\mathbf{C} \times \mathbf{M} \times \mathbf{K} \to \mathbf{W}$, $w = U(c, m, k)$. Тобто вважається за доцільне враховувати як особливості вбудовуваного повідомлення, так і особливості контейнера, до якого його планується ввести.

Довільність функції U обмежується вимогами стійкості до різного роду впливів на отриманий контейнер-результат. Крім того, функція U є складеною:

$$U = T \circ G, \tag{2.4}$$

де G: $\mathbf{M} \times \mathbf{K} \to \mathbf{Z}$; T: $\mathbf{C} \times \mathbf{Z} \to \mathbf{W}$.

Функція G може бути реалізована, наприклад, за допомогою криптографічного безпечного генератора ПВП з ключем $\mathbf{K}$ в якості початкового значення. Для підвищення стійкості стеганограми можуть застосовуватися завадостійкі коди, наприклад, коди Геммінґа, БЧХ, Ґолея, згорткові коди [65, 115]. Оператор T модифікує кодові слова $\mathbf{Z}$ з урахуванням формату контейнера, в результаті чого одержується оптимальне для вбудовування повідомлення. Функція T повинна бути обрана таким чином, щоб контейнер-оригінал $\mathbf{C}$, контейнер-результат $\mathbf{S}$ і модифікований у передбачених межах контейнер-результат $\hat{\mathbf{S}}$ породжували одне й те саме оптимальне для вбудовування повідомлення:

$$T: \mathbf{C} \times \mathbf{Z} = T: \mathbf{S} \times \mathbf{Z} = T: \hat{\mathbf{S}} \times \mathbf{Z} = \to W. \tag{2.5}$$

Процес вбудовування повідомлення $\mathbf{W}$ до контейнера-оригіналу $\mathbf{C}$ при цьому можна описати як суперпозицію сигналів:

$$E: \mathbf{C} \times \mathbf{V} \times \mathbf{W} \to \mathbf{S}; \quad s(x, y) = c(x, y) * v(x, y)\, w(x, y)\, p(x, y), \tag{2.6}$$

де $v(x, y)$ — маска вбудовування повідомлень, що враховує характеристики відповідної системи сприйняття середньостатистичної людини і слугує для зменшення помітності (відчутності) цих повідомлень; $p(x, y)$ — проектуюча, залежна від ключа функція; знак «*» позначує оператор суперпозиції, що в загальному випадку включає до

себе, окрім додавання, обмеження рівня (або нормування) і квантування.

Проектуюча функція здійснює «розподіл» оптимізованого повідомлення по всій області контейнера. Її використання може розглядатися як реалізація рознесення конфіденційної інформації паралельними каналами. Крім того, дана функція має певну просторову структуру і кореляційні властивості, що використовуються для протидії, наприклад, деяким типам атак (див. розділ 3).

Ще один можливий опис процесу вбудовування наведено у [5] з посиланням на [41]. Представимо стеганографічну систему як систему зв'язку з передаванням додаткової інформації (рис. 2.4). У цій моделі кодер і декодер мають доступ, окрім ключа, ще й до інформації про канал (тобто про контейнер і про можливі атаки). У залежності від положення перемикачів ***А*** і ***Б*** виділяють чотири класи стеганосистем (при цьому вважається, що ключ завжди відомий як кодеру, так і декодеру).

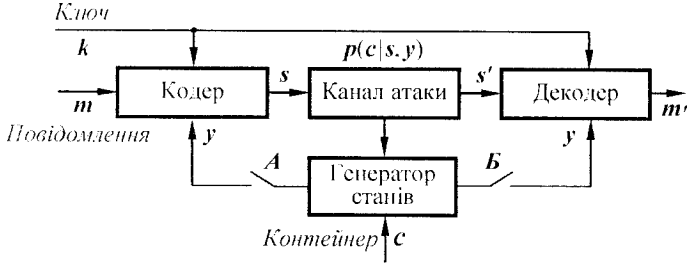


Рис. 2.4. Представлення стеганосистеми як системи зв'язку з передаванням додаткової інформації

- **I клас.** Додаткова інформація відсутня (перемикачі розімкнуті). Це так звані «класичні» стеганосистеми. У ранніх роботах зі стеганографії вважалося, що інформація про канал є недоступною для кодека. Виявлення прихованої інформації здійснювалося шляхом обчислення коефіцієнта кореляції між прийнятим контейнером і обчисленим за ключем повідомленням Якщо коефіцієнт перевищував деякий поріг, приймалося рішення щодо присутності вбудованих даних. Але відомо, що кореляційний приймач є оптимальним лише у випадку адитивної гаусівської завади [115]. За інших типів атак (наприклад, геометричних спотвореннях) дані стеганосистеми давали незадовільні результати.
- **II клас.** Інформація про канал є відомою тільки кодерові (ключ ***А*** замкнутий, ***Б*** розімкнутий). Така конструкція привернула до себе

увагу завдяки роботі [47]. Особливістю схеми є те, що, працюючи «всліпу», вона має ту ж теоретичну пропускну здатність, що й схема з наявністю контейнера-оригіналу в декодері. До недоліків стеганосистем класу II можна віднести високу складність кодера (необхідність побудови кодової книги для кожного контейнера), а також відсутність адаптації схеми до можливих атак. Останнім часом запропоновано низку практичних підходів, які усувають ці недоліки. Зокрема, для зниження складності кодера пропонується використовувати структуровані кодові книги, а декодер розраховувати на випадок найгіршої атаки.

- **III клас.** Додаткова інформація є відомою тільки декодеру (перемикач ***А*** розімкнутий, ***Б*** замкнутий). Декодер будується з урахуванням можливих атак. У результаті одержуються стійкі до геометричних атак стеганосистеми. Одним з методів досягнення цієї мети є використання так званого опорного вбудованого повідомлення (аналог пілот-сигналу в радіозв'язку). Опорне повідомлення — невелика кількість бітів, вбудовуваних в інваріантні до перетворювань коефіцієнти сигналу. Наприклад, можна виконати вбудовування в амплітудні коефіцієнти перетворення Фур'є, які є інваріантними до афінних (геометричних) перетворювань. Тоді опорне повідомлення вказуватиме, яке перетворення виконав над контейнером порушник. Іншим призначенням пілотного повідомлення є боротьба із завмираннями, за аналогією з радіозв'язком. Завмираннями в даному контексті можна вважати зміну значень відліків сигналу при вбудовуванні даних, атаках, додаванні негаусівського шуму тощо. У радіозв'язку для боротьби із завмираннями використовується метод рознесеного прийому (по частоті, у часі чи просторі, за кодом). У стеганографії ж використовується рознесення вбудованих повідомлень у просторі контейнера. Пілотне повідомлення генерується в декодері на основі ключа.
- **IV клас.** Додаткова інформація є відомою як у кодері, так і в декодері (обидва перемикачі замкнуті). Як відзначено у [46], всі перспективні стеганосистеми повинні будуватися саме за цим принципом. Оптимальність такої схеми досягається шляхом оптимального узгодження кодера з сигналом-контейнером, а також адаптивним управлінням декодером в умовах спостереження за каналом атак.

2.5. Підсумки розділу

У даному розділі шляхом аналізу спеціалізованих літературних джерел та ресурсів мережі *Internet* наведено узагальнене визначення поняття стеганографічної системи, визначено існуючі та перспективні напрямки, за якими можливе використання стеганографії як інструменту захисту інформації в автоматизованих системах, окреслено проблему співвідношення між стійкістю стеганосистеми та об'ємом прихованого за її допомогою повідомлення, розкрито сутність таких основних понять стеганографії як повідомлення, контейнер-оригінал, контейнер-результат, стеганоключ, стеганоканал та ін.

Усе це дозволило безпосередньо перейти до побудови структурної схеми стеганосистеми, яку проведено з позицій теорії зв'язку, виконати систематизований огляд відомих протоколів стеганосистем. Отримані результати дозволяють зробити висновок про недоцільність використання безключових стеганосистем, безпека яких базується лише на таємності використовуваних стеганографічних перетворень.

Наведено переваги використання відкритого ключа порівняно із секретним. До основного переліку стеганографічних протоколів запропоновано включити протокол, що розуміє під собою додаткове попереднє перетворення прихованої інформації до оптимального формату, виходячи з особливостей формату носія, який планується використати в якості контейнера.

Розділ 3. ПРИНЦИПИ СТЕГАНОГРАФІЧНОГО АНАЛІЗУ

3.1. Вступні положення

Основною метою стеганоаналізу є моделювання стеганографічних систем та їхнє дослідження для отримання якісних і кількісних оцінок надійності використовуваного стеганоперетворення, а також побудова методів виявлення прихованої в контейнері інформації, її модифікації чи руйнування.

Термінологія стеганоаналізу значною мірою є аналогічною термінології криптоаналізу, однак присутні й певні розбіжності. Зокрема, криптоаналіз застосовується з метою дешифрування змісту криптограм, а стеганоаналіз — для виявлення наявності прихованої інформації.

За рівнем забезпечення таємності стеганосистеми поділяються на теоретично стійкі, практично стійкі і нестійкі системи [3].

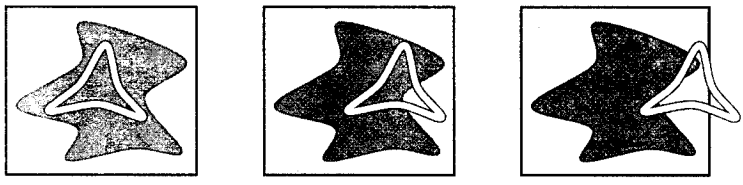
Теоретично стійка (абсолютно надійна) стеганографічна система здійснює приховання інформації лише в тих фрагментах контейнера, значення елементів яких не перевищують рівень шумів або помилок квантування, і при цьому теоретично доведено, що неможливо створити стеганоаналітичний метод виявлення прихованої інформації (рис. 3.1, *а*).

Практично стійка стеганосистема здійснює таку модифікацію фрагментів контейнера, зміни яких можуть бути виявлені, але відомо, що на даний момент необхідні стеганоаналітичні методи в порушника відсутні або ж узагалі поки що не розроблені (рис. 3.1, *б*).

Нестійка стеганосистема приховує повідомлення в такий спосіб, що існуючі стеганоаналітичні засоби дозволяють з легкістю його виявити (рис. 3.1, *в*). У цьому випадку стеганографічний аналіз допомагає знайти уразливі місця стеганографічного перетворення і провести його удосконалення таким чином, щоб усі зміни, внесені до контейнера, знову виявилися би в області теоретичної або, принаймні, практичної нерозрізненості (рис. 3.1, *а*/*б*).

Порушник може бути пасивним, активним і зловмисним. Залежно від цього він може створювати й різні рівні загроз. *Пасивний порушник* може лише виявити факт наявності стеганоканалу і, можливо, дізнаватися про вміст повідомлення. Чи буде він здатен прочитати повідомлення після його виявлення залежить від стійкості системи шифрування, і це питання, як правило, у стеганографії не розглядається. Діапазон дій *активного порушника* є значно ширшим. Приховане повідомлення може

бути ним не лише виявлене, але й навіть видалене чи пошкоджене. Дії *зловмисного порушника* найбільш небезпечні. Він здатен не лише зруйнувати, але й створювати фальшиві стеганограми (дезінформацію) [40]. Аби здійснити той чи інший тип загрози порушник застосовує атаки.



а) *б)* *в)*

– область *практичної* нерозрізненості, в якій зміни контейнера не виявляються існуючими у порушника аналітичними методами;

– область *теоретичної* нерозрізненості, в якій приховані елементи практично не можуть бути виявлені, оскільки перебувають нижче рівня шумів і помилок квантування;

– область захисту стеганографічної системи.

Рис. 3.1. Співвідношення методів стеганозахисту і стеганоаналізу:
а — теоретично стійка стеганосистема;
б — практично стійка стеганосистема;
в — нестійка стеганосистема.

3.2. Види атак на стеганографічні системи

Стеганосистема вважається *зламаною*, якщо порушник, перехопивши контейнер, зумів, принаймні, довести існування у ньому прихованого повідомлення. Передбачається, що порушник здатний проводити будь-які види атак і має необмежені обчислювальні ресурси. Якщо йому не вдається підтвердити гіпотезу про те, що в контейнері приховано секретне повідомлення, то стеганографічна система вважається *стійкою*.

У більшості практичних випадків виділяють декілька етапів зламу стеганографічної системи [40]:

- виявлення факту присутності прихованої інформації;
- видобування прихованого повідомлення;
- видозміна (модифікація) прихованої інформації;
- заборона на здійснення будь-якого пересилання інформації, у тому числі і прихованої.

Перші два етапи відносяться до *пасивних* атак на стеганосистему, а останні два — до *активних* (або так званих *зловмисних*) атак. За аналогією з криптоаналізом виділяють наступні види атак на стеганосистеми [3, 5]:

- *атака на основі відомого заповненого контейнера.* У цьому випадку порушник має у своєму розпорядженні один або декілька перехоплених заповнених контейнерів (в останньому випадку передбачається, що вбудовування прихованої інформації здійснювалося однаковим способом). Завдання порушника може складатися у виявленні факту наявності стеганоканалу (основне завдання), а також у видобуванні даних чи визначенні ключа. Знаючи ключ, порушник матиме можливість аналізу інших стеганограм;
- *атака на основі відомого вбудованого повідомлення.* Цей тип атаки більшою мірою характерний для систем захисту інтелектуальної власності, коли в якості, наприклад, ЦВЗ використовується відомий логотип фірми. Завданням аналізу є одержання ключа. Якщо відповідний прихованому повідомленню заповнений контейнер невідомий, то таке завдання є практично не розв'язним;
- *атака на основі обраного прихованого повідомлення.* У цьому випадку порушник може пропонувати для передавання свої повідомлення й аналізувати отримувані при цьому контейнери-результати;
- *адаптивна атака на основі обраного повідомлення.* Ця атака є окремим випадком попередньої. При цьому порушник має можливість обирати повідомлення для нав'язування їх адаптивно, у залежності від результатів аналізу попередніх контейнерів-результатів;
- *атака на основі обраного заповненого контейнера.* Цей тип атаки є більше характерним для систем з ЦВЗ. Стеганоаналітик має детектор заповнених контейнерів у вигляді «чорної скриньки» і декілька таких контейнерів. Аналізуючи продетектовані стеганограми, порушник намагається розкрити ключ.

Крім того, в порушника може існувати можливість застосувати ще три типи атак, що не мають прямих аналогій у криптоаналізі:

- *атака на основі відомого порожнього контейнера.* Якщо останній є відомим порушнику, то шляхом порівняння його з підозрюваним на присутність прихованих даних контейнером, він завжди може встановити факт наявності стеганоканалу. Незважаючи на тривіальність цього випадку, у ряді робіт приводиться його теоретико-інформаційне обґрунтування. Набагато цікавішим ба-

читься сценарій, коли контейнер відомий приблизно, з деякою похибкою (як це може мати місце при додаванні до нього шуму). У цьому випадку існує можливість побудови стійкої стеганосистеми [5];

- *атака на основі обраного порожнього контейнера.* У цьому випадку порушник намагається змусити користуватися запропонованим ним контейнером. Останній, наприклад, може мати значні однорідні області (однотонні зображення, класична музика тощо), і тоді буде важко забезпечити таємність вбудовування;
- *атака на основі відомої математичної моделі контейнера або його частини.* При цьому атакуючий намагається визначити відмінність підозрілих медіа-даних від відомої йому моделі. Наприклад, можна припустити, що біти всередині певної частини зображення є корельованими. Тоді відсутність такої кореляції може служити сигналом про наявне приховане повідомлення. *Завдання того, хто вбудовує повідомлення, полягає у тому, щоб* не порушити наявну статистику контейнера. Відправник і зловмисник можуть мати у своєму розпорядженні різні моделі сигналів, тоді в інформаційно-приховуючому протиборстві переможе той, хто має ефективнішу (більш оптимальну) модель.

Основна мета атаки на стеганографічну систему є аналогічною атакам на криптосистему, з тією лише різницею, що у першому випадку різко зростає значимість активних (зловмисних) атак. Будь-який контейнер може бути замінений з метою видалення або руйнування прихованого повідомлення, незалежно від того, існує воно насправді в контейнері чи ні. Виявлення існування прихованих даних зберігає час на етапі їхнього видалення, тому що буде потрібно обробляти тільки ті контейнери, які містять приховану інформацію.

Навіть за найкращих умов для атаки, задача видобування прихованого повідомлення з контейнера може виявитися надскладною. Однозначно стверджувати про факт існування прихованої інформації можна лише після її виділення в явному вигляді. Іноді метою стеганографічного аналізу є не алгоритм взагалі, а пошук, наприклад, конкретного стеганоключа, що використовується для вибору бітів контейнера під час стеганоперетворення.

3.3. Основні етапи практичного стеганоаналізу

Фактично, будь-яке стеганографічне перетворення базується на двох визначальних принципах [3]:

- в якості носія прихованої інформації (контейнера) обирається той об'єкт, структура якого припускає можливість певного спотворення власної інформації, зберігаючи при цьому його колишню функціональність;
- рівень внесених до структури контейнера спотворень має бути нижчим за рівень чутливості засобів розпізнавання (у тому числі — й за рівень розпізнавання органами відчуття людини).

Як вже зазначалося, в якості стеганоконтейнерів можуть використовуватися майже всі відомі медіа-дані, застосовувані у сучасних інфокомунікаційних системах. При цьому методи приховання інформації орієнтуються в основному на внутрішню структуру контейнера, що може бути представлена символьними або бітовими даними, коефіцієнтами ортогональних перетворень, широкосмуговим кодуванням, коефіцієнтами ущільнення тощо [3].

Приховання даних у медіасередовищі вимагає дотримання певних умов при внесенні змін, що має на меті усунення прояву слідів застосування операцій стеганоперетворення. Так, наприклад, у випадку використання зображень зазначені зміни можуть за певних дій з боку порушника (як навмисних, так і випадкового характеру) ставати видимими для людського ока і, отже, явно вказувати на застосування стеганографічних перетворень. Очевидно, що сліди, залишені останніми, можуть істотно допомогти у детектуванні прихованого повідомлення, компрометуючи, у підсумку, стеганосистему в цілому.

Однією з головних *задач стеганоаналізу* є дослідження можливих слідів застосування стеганографічних засобів і розробка методів, які б дозволяли виявити факти їхнього використання [3]. Застосування конкретного стеганографічного перетворення вимагає від стеганоаналітика індивідуального підходу до його дослідження.

Дослідження повідомлень, прихованих одним з множини існуючих стеганографічних методів, або, більш точно, підозрюваних у цьому відношенні, є досить трудомістким процесом.

Для успішного проведення стеганоаналізу є необхідним, але в жодному разі не достатнім:

- мати для аналізу стеганографічний засіб, за допомогою якого здійснюється приховання повідомлення;
- мати можливість відновлювати застосовувані в системі стеганографічний і, можливо, криптографічний алгоритми, проводити їхній експертний аналіз і розробляти алгоритм визначення ключів;
- мати змогу використовувати для проведення стеганоаналізу обчислювальний ресурс достатньої потужності;
- підтримувати на належному рівні теоретичні і практичні знання в галузі комп'ютерної стеганографії.

Можна виділити наступні декілька напрямків практичного розвитку стеганографічного аналізу [3]:

- Розробка ймовірнісно-статистичних методів розпізнавання, застосування елементів штучного інтелекту для отримання оцінок надійності стеганографічних перетворень, а також при створенні детекторів (фільтрів) для аналізу інформаційних потоків з метою виявлення і перекриття прихованих каналів зв'язку. У такому випадку перевірка наявності прихованої інформації зводиться до певної оцінки з використанням статистичних критеріїв (послідовної кореляції, ентропії зображення, дисперсії молодшого біта тощо). Розроблювані з цією метою засоби повинні не тільки забезпечувати низький рівень похибки під час розпізнавання прихованих повідомлень (особливо в тих випадках, коли використовується попереднє шифрування), але й бути універсальними, тобто повинна існувати можливість детектування повідомлень вбудованих різними стеганографічними методами.
- Аналіз конкретних програмних стеганографічних засобів з метою відновлення алгоритмів і оптимальної розробки методу їхнього дослідження. Основна складність в даному випадку полягає у великій трудомісткості, зумовленій необхідністю індивідуального підходу до кожного конкретного алгоритму, який реалізує метод приховання інформації, а також значним об'ємом обчислень, необхідних для відновлення стеганоключів.
- Розробка технологій активних і зловмисних атак для внесення невідновлюваних спотворень у гіпотетичну стеганограму з метою спровокувати її повторне передавання в іншому контейнері, що підтвердило б факт використання стеганозасобів.

3.4. Оцінювання якості стеганосистеми

Створення й експлуатація надійного стеганографічного засобу передбачає наявність визначеного інструментарію для його контролю й оцінювання [3]. Кількісне оцінювання стійкості стеганографічної системи захисту до зовнішніх впливів являє собою досить складну задачу, яка на практиці зазвичай реалізується методами системного аналізу, математичного моделювання або експериментального дослідження.

Як правило, професійно розроблена стеганосистема забезпечує трирівневу модель захисту інформації, що вирішує такі основні задачі:

- приховання самого факту наявності інформації, що захищається;

- блокування несанкціонованого доступу до інформації, що здійснюється шляхом обрання відповідного методу приховання інформації;
- попередній криптографічний захист (шифрування) прихованої інформації.

На рис. 3.2 представлена можлива структура процесу моделювання й оцінювання стійкості стеганосистеми [3, 48]. Як видно, надійність і час стійкості стеганосистеми у випадку проведення аналізу й випробувань визначаються обчислювальними можливостями комплексу.

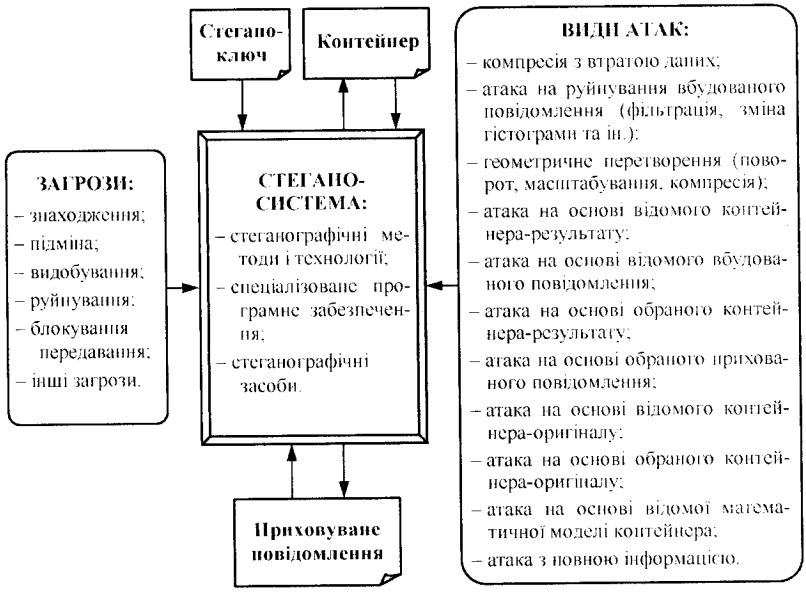


Рис. 3.2. Модель аналізу загроз й оцінювання стійкості стеганосистем

Оцінювання якості основної характеристики стеганосистеми — *рівня прихованості* — забезпечується шляхом проведення аналітичних досліджень (стеганоаналізу) і натурних випробувань [3]. Для оцінювання якості стеганографічного приховання часто застосовують відомі методи з інших областей, у першу чергу — криптоаналізу. Оскільки абонент-одержувач може відновлювати приховану інформацію з прийнятого повідомлення, то цілком очевидно, що існує і деякий механізм її видобування. Якщо порушник, висуваючи гіпотези про можливе стеганографічне перетворення, має певний інструмент для їхньої перевірки, то він має шанси на підтвердження чи спростування

факту існування прихованої інформації, здійснення пошуку механізму видобування секретного повідомлення і, зрештою, розкриття змісту повідомлення. Тому, у першу чергу, для детектування стеганограм можливе застосування різновидів описаних вище атак на стеганосистему і значну частину методів криптоаналізу.

У деяких випадках досить ефективними можуть виявитися методи оцінювання рівня прихованості стеганозасобів на основі аналізу їхніх *статистичних характеристик*. Статистична теорія дає кількісні критерії випадковості, що дозволяє створювати детектори, які б виявляли статистичні розбіжності між послідовностями. У випадку наявності необхідного обсягу проведених стеганоаналізів, з достатньо високою імовірністю можна робити і висновки про основні характеристики послідовності, виділеної для аналізу з контейнера. При цьому на початковому етапі рекомендується скористатися традиційними *статистичними* (χ^2, тести на заборонені символи, на довжину циклу тощо), *емпіричними* (перевірки частот, серій, інтервалів, перестановок; перевірки на монотонність, «покер-тест», тест «збирача купонів») або *спектральними* тестами [3]. А вже у подальшому доцільно використовувати більш гнучкі методи, які іноді спеціально розроблюються під кожну конкретну задачу.

Для порівняльного оцінювання якості стеганографічних засобів розроблюють різні *показники*, що повертають кількісні оцінки. Найбільша їхня кількість розроблена для СА, які працюють із зображеннями та відео (особливо — вбудовування ЦВЗ). Зазвичай такі показники оперують із зображенням на рівні пікселів, хоча після належної адаптації вони застосовні й до інших способів опису зображення, а також — і до аудіопослідовностей. Найбільш популярним показником при аналізі рівня спотворювань, які вносяться у контейнер під час приховання в ньому інформації, є взяте з радіотехніки обчислюване у децибелах співвідношення «сигнал/шум».

Низку показників, використовуваних під час оцінювання спотворень, що вносяться стеганоперетвореннями до зображення, наведено у табл. 3.1 [49–52].

Більшість показників спотворення або критерії якості, що використовуються при візуальній обробці інформації, належать до групи *різницевих*. Дані показники базуються на відмінностях між контейнером-оригіналом (неспотворений сигнал) і контейнером-результатом (спотворений сигнал).

До другої групи входять показники, засновані на кореляції між оригінальним і спотвореним сигналами (так звані *кореляційні показники спотворення*).

Поширені показники візуального спотворення, засновані на аналізі піксельної структури контейнера

Різницеві показники спотворення	
Максимальна різниця *(Maximum Difference)*	$$MD = \max_{x,y} \left\| \mathbf{C}_{x,y} - \mathbf{S}_{x,y} \right\|. \quad (3.1)$$
Середня абсолютна різниця *(Average Absolute Difference)*	$$AD = \frac{1}{XY} \cdot \sum_{x,y} \left\| \mathbf{C}_{x,y} - \mathbf{S}_{x,y} \right\|. \quad (3.2)$$
Нормована середня абсолютна різниця *(Normalized Average Absolute Difference)*	$$NAD = \sum_{x,y} \left\| \mathbf{C}_{x,y} - \mathbf{S}_{x,y} \right\| \Big/ \sum_{x,y} \left\| \mathbf{C}_{x,y} \right\|. \quad (3.3)$$
Середньоквадратична помилка *(Mean Square Error)*	$$MSE = \frac{1}{XY} \cdot \sum_{x,y} \left(\mathbf{C}_{x,y} - \mathbf{S}_{x,y} \right)^2. \quad (3.4)$$
Нормована середньоквадратична помилка *(Normalized Mean Square Error)*	$$NMSE = \sum_{x,y} \left(\mathbf{C}_{x,y} - \mathbf{S}_{x,y} \right)^2 \Big/ \sum_{x,y} \left(\mathbf{C}_{x,y} \right)^2. \quad (3.5)$$
L^p-норма *(L^p-norm)*	$$L^p = \left(\frac{1}{XY} \cdot \sum_{x,y} \left\| \mathbf{C}_{x,y} - \mathbf{S}_{x,y} \right\|^p \right)^{1/p}. \quad (3.6)$$
Лапласова середньоквадратична помилка *(Laplacian Mean Square Error)**	$$LMSE = \sum_{x,y} \left(\nabla^2 \mathbf{C}_{x,y} - \nabla^2 \mathbf{S}_{x,y} \right)^2 \Big/ \sum_{x,y} \left(\nabla^2 \mathbf{C}_{x,y} \right)^2, \quad (3.7)$$ де $\nabla^2 \mathbf{C}_{x,y} = \mathbf{C}_{x+1,y} + \mathbf{C}_{x-1,y} + \mathbf{C}_{x,y+1} + \mathbf{C}_{x,y-1} - 4 \cdot \mathbf{C}_{x,y}$.
Відношення сигнал/шум *(Signal to Noise Ratio)*	$$SNR = \sum_{x,y} \left(\mathbf{C}_{x,y} \right)^2 \Big/ \sum_{x,y} \left(\mathbf{C}_{x,y} - \mathbf{S}_{x,y} \right)^2. \quad (3.8)$$
Максимальне відношення сигнал/шум *(Peak Signal to Noise Ratio)*	$$PSNR = XY \cdot \max_{x,y} \left(\mathbf{C}_{x,y} \right)^2 \Big/ \sum_{x,y} \left(\mathbf{C}_{x,y} - \mathbf{S}_{x,y} \right)^2. \quad (3.9)$$
Якість зображення *(Image Fidelity)*	$$IF = 1 - \sum_{x,y} \left(\mathbf{C}_{x,y} - \mathbf{S}_{x,y} \right)^2 \Big/ \sum_{x,y} \left(\mathbf{C}_{x,y} \right)^2. \quad (3.10)$$

Кореляційні показники спотворення	
Нормована взаємна кореляція *(Normalized Cross-Correlation)*	$$NC = \sum_{x,y} \mathbf{C}_{x,y} \cdot \mathbf{S}_{x,y} \Big/ \sum_{x,y} \left(\mathbf{C}_{x,y}\right)^2. \quad (3.11)$$
Якість кореляції *(Correlation Quality)*	$$CQ = \sum_{x,y} \mathbf{C}_{x,y} \cdot \mathbf{S}_{x,y} \Big/ \sum_{x,y} \mathbf{C}_{x,y}. \quad (3.12)$$
Інші показники	
Структурний зміст *(Structural Content)*	$$SC = \sum_{x,y} \left(\mathbf{C}_{x,y}\right)^2 \Big/ \sum_{x,y} \left(\mathbf{S}_{x,y}\right)^2. \quad (3.13)$$
Загальне сигма-відношення сигнал/шум *(Global Sigma Signal to Noise Ratio)*	$$GSSNR = \sum_b \sigma_b^2 \Big/ \sum_b \left(\sigma_b - \tilde{\sigma}_b\right)^2, \quad (3.14)$$ де $\sigma_b = \sqrt{\frac{1}{n} \cdot \sum_{\text{блок } b} \left(\mathbf{C}_{x,y}\right)^2 - \left(\frac{1}{n} \cdot \sum_{\text{блок } b} \mathbf{C}_{x,y}\right)^2}$.
Сигма-відношення сигнал/шум *(Sigma Signal to Noise Ratio)*	$$SSNR' = \frac{1}{n} \cdot \sum_b SSNR_b, \quad (3.15)$$ де $SSNR_b = 10 \cdot lg\left[\frac{\sigma_b^2}{\left(\sigma_b - \tilde{\sigma}_b\right)^2}\right]$.
Нормоване відношення сигма/помилка *(Normalized Sigma to Error Ratio)*	$$NSER = \frac{1}{\max(SER)} \cdot \sum_b SER_b, \quad (3.16)$$ де $SER_b = \sigma_b^2 \Big/ \left[\frac{1}{n} \cdot \sum_{\text{блок } b} \left(\mathbf{C}_{x,y} - \mathbf{S}_{x,y}\right)^2\right]$.
Подібність гістограм *(Histogram Similarity)*	$$HS = \sum_{c=0}^{255} \left\lvert f_C(c) - f_S(c) \right\rvert, \quad (3.17)$$ де $fc(c)$ — відносна частота градації кольору c у зображенні з 256 рівнями кольорів.

У наведених співвідношеннях через $\mathbf{C}_{x,y}$ позначається піксель порожнього контейнера з координатами (x, y), а через $\mathbf{S}_{x,y}$ — відповідний піксель заповненого контейнера. У параметрах ***GSSNR***, ***SSNR*** та ***SER*** аналізоване зображення попередньо розбивається на N блоків по n пікселів розміром $X \times Y$, де X і Y — кількість рядків і стовпців у блоці відповідно (наприклад, блок 8×8 пікселів). Більш детальний опис показників можна отримати, зокрема, з [52].

Розглянуті показники базуються на аналізі окремих елементів сигналу (у даному випадку — пікселів зображення), тому їхні слабкі місця у контексті застосування в стеганографії очевидні: відсутність корельованості із зором людини. В останнє десятиріччя все більше досліджень спрямовано на винайдення такого показника спотворення, який би був адаптований до людської зорової чи слухової системи шляхом врахування різноманітних впливів [53–56]. Розглянемо показник спотворення, який запропонували свого часу Джойс Фаррелл *(J. E. Farrell)* та Крістіан ван ден Бранден Ламбрехт *(C. J. van den Branden Lambrecht)* [55].

Ступінь сприйманої людиною якості оперує чутливістю до контрасту та явищем маскування системою візуалізації людини і базується на багатоканальній моделі людського просторового зору.

Обчислення даного показника вимагає наступних дій:

- проведення великокрокової сегментації зображення;
- розкладання помилки кодування і первинного зображення на перцепційні (такі що відносяться до сприйняття органами відчуттів) компоненти, використовуючи гребінчасті фільтри;
- обчислення порогу виявлення для кожного пікселя, використовуючи первинне зображення в якості маски;
- розподілення фільтрованої помилки за допомогою порогу прийняття рішення;
- об'єднання отриманих результатів по всім колірним каналам.

Одиниця вимірювання показника визначається як одиниця *перевищення порогу*, що розуміє під собою тільки значущу (відчутну) відмінність (*JND – Just Noticeable Difference*). Загальним показником є приховане максимальне відношення сигнал/шум (*MPSNR – Masked Peak Signal to Noise Ratio*):

$$\boldsymbol{MPSNR} = 10 \cdot \boldsymbol{lg}\left(\frac{255^2}{\boldsymbol{\varepsilon}^2}\right), \tag{3.18}$$

де $\boldsymbol{\varepsilon}$ — обчислене спотворення.

Оскільки даний показник якості не в повній мірі відповідає сенсу, закладеному у поняття «децибел», його називають *візуальним* (або *зоровим*) *децибелом* (ВдБ).

У більшості випадків більш корисною є *нормалізована оцінка якості*. У [49] пропонується використовувати оцінку якості $\boldsymbol{Q}$ у відповідності до рекомендацій *ITU-R Rec. 500* сектора радіозв'язку МСЕ:

$$\boldsymbol{Q} = 5/(\boldsymbol{N} \cdot \boldsymbol{\varepsilon} + 1), \tag{3.19}$$

де $\boldsymbol{\varepsilon}$ — обчислене спотворення; $\boldsymbol{N}$ — нормувальний коефіцієнт, що

зазвичай обирається таким, аби характеристика спотворення відображувала собою відповідну якісну оцінку.

Така оцінка має декілька переваг, зокрема — відсутність руйнування неспотворених зображень. Оцінки і відповідні їм зорове сприйняття та якість наведені у табл. 3.2.

Таблиця 3.2

ITU-R Rec. 500. Оцінки якості за шкалою від 1 до 5

Оцінка	*Спотворення*	*Якість*
«5»	Непомітне	Відмінна
«4»	Помітне, але не подразнююче	Добра
«3»	Несуттєво подразнююче	Задовільна
«2»	Подразнююче	Незадовільна
«1»	Надзвичайно подразнююче	Вкрай незадовільна

3.5. Абсолютно надійна стеганосистема

У [3, 15] наведено формальне теоретико-інформаційне визначення стійкості стеганосистеми до пасивних атак, головна ідея якого базується на випадковому характеру обрання контейнера c з множини $\mathbf{C}$ — з імовірністю P_C.

Вбудовування до контейнера секретного повідомлення можна описати як функцію, визначену на множині $\mathbf{C}$. Нехай P_S — імовірність формування контейнера-результату $s = E(c, m, k)$ на множині $\mathbf{S}$ усіх можливих контейнерів, отриманих за допомогою певної стеганосистеми. Якщо контейнер c ніколи не використовується для отримання стеганограми, то $P_S(c) = 0$. Для обчислення імовірності P_S необхідно врахувати розподіл імовірностей на множині ключів $\mathbf{K}$ та множині повідомлень $\mathbf{M}$.

Визначимо на множині Q таке співвідношення для *відносної ентропії*, за допомогою якого можна виміряти неефективність прийняття хибної гіпотези про розподіл P_1 у випадку істинного розподілу P_0:

$$D\left(P_0 \| P_1\right) = \sum_{q \in Q} P_0(q) \cdot \log_2\left(\frac{P_0(q)}{P_1(q)}\right), \qquad (3.20)$$

де вираз $\log_2(\bullet)$ являє собою логарифмічне відношення правдоподібності.

Відносна ентропія між двома розподілами завжди є невід'ємною і дорівнює 0 лише у випадку тотожності даних розподілів. Таким чином, для стеганоперетворення можна отримати деяку оцінку. Наведемо визначення надійності стеганосистеми у термінах відносної ентропії.

Означення 3.1

Нехай Σ — стеганографічна система; P_S — розподіл імовірностей передавання каналом зв'язку стеганограм; P_C — розподіл імовірностей передавання каналом зв'язку пустих контейнерів. Система Σ називається *ρ-надійною* до пасивних атак, якщо виконується нерівність $D\left(P_C \| P_S\right) \le \rho$, і є *абсолютно надійною*, якщо $\rho = 0$.

Як вже зазначалося, співвідношення $D\left(P_C \| P_S\right)$ дорівнює нулеві тільки у тому випадку, коли обидва розподіли імовірностей дорівнюють один одному. Отже, стеганосистема Σ є *теоретично абсолютно надійною*, якщо процес вбудовування секретного повідомлення до контейнера не змінює розподіл P_C. Абсолютно безпечна стеганосистема може бути створена, наприклад, на основі одноразової гами (за аналогією з потоковими шифрами гамування) [3].

На підставі сказаного, формулюється наступна теорема.

Теорема 3.1

Існує абсолютно надійна стеганосистема.

Доведення.

У [15] здійснене конструктивне доведення даного твердження. Нехай контейнер c являє собою рівномірно розподілену n-бітову послідовність для деякого додатного n. Відправник за допомогою генератора ключів одержує рівномірно розподілений n-бітовий ключ k. Вважається, що функція вбудовування полягає у побітовому складанні за модулем 2 (операція *XOR*) n-бітового секретного повідомлення (в ролі якого в даному випадку і виступає контейнер c) з ключем k: $s = c \oplus k$. Одержувач декодує отриману послідовність повторним застосуванням операції *XOR*: $c = s \oplus k$. Є цілком очевидним, що результуючий контейнер s також являтиме собою рівномірно розподілену n-бітову послідовність. Отже, $P_C \sim P_S$, звідки $D\left(P_C \| P_S\right) = 0$.

Теорему доведено.

3.6. Стійкість стеганосистем до пасивних атак

Пасивний порушник намагається знайти відповідь на питання — містить перехоплений ним контейнер приховану інформацію чи ні? Для цього йому необхідно провести оцінювання неоднорідності деяких параметрів контейнера, виявити в ньому «підозрілі» ділянки, завищене зашумлення або якісь інші сліди присутності прихованих повідомлень. Ця задача може бути формалізована у вигляді проблеми *перевірки статистичних гіпотез* [5]. З цією метою вводять тестову функцію стеганодетектора, яка, в залежності від типу останнього, може видавати дворозрядні (а в більш складному випадку — μ-розрядні) результати про наявність/відсутність вбудованого повідомлення:

$$\boldsymbol{D}: \mathbf{C} \rightarrow \{0; 1\}, \tag{3.21}$$

$$\boldsymbol{D}(c) = \begin{cases} 1, \text{ якщо контейнер містить приховане повідомлення;} \\ 0, \text{ якщо приховане повідомлення в контейнері відсутнє.} \end{cases}$$

За допомогою даної функції порушник здатен оцінювати повідомлення, перехоплені ним у відкритому каналі. В якості детектора прихованих повідомлень зазвичай використовують кореляційний приймач [115], зображений на рис. 3.3.

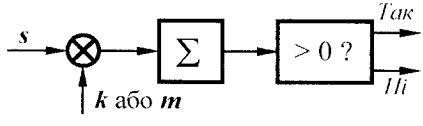


Рис. 3.3. Кореляційний детектор прихованих повідомлень

Нехай внаслідок приховання у зображенні-контейнері повідомлення, в деякої частини пікселів значення яскравості було збільшено на 1, а в інших — лишилося незмінним або ж було зменшене на 1. Тоді $\boldsymbol{s} = \boldsymbol{c} + \boldsymbol{m}$, де $\boldsymbol{m} = \boldsymbol{E}(\boldsymbol{c}, \boldsymbol{k})$. Корелятор детектора обчислює величину

$$\boldsymbol{s}\cdot\boldsymbol{m} = (\boldsymbol{c} + \boldsymbol{m})\cdot\boldsymbol{m} = \boldsymbol{c}\cdot\boldsymbol{m} + \boldsymbol{m}\cdot\boldsymbol{m}.$$

Оскільки $\boldsymbol{m}$ приймає значення ± 1, то $\sum \boldsymbol{c}\cdot\boldsymbol{m} \rightarrow 0$, а $\boldsymbol{m}\cdot\boldsymbol{m}$ завжди буде додатнім. Тому $\boldsymbol{s}\cdot\boldsymbol{m}$ буде дуже близьким до $\boldsymbol{m}\cdot\boldsymbol{m}$. Тоді можна скористатися відомостями з теорії зв'язку і записати ймовірність помилкового виявлення наявності прихованого повідомлення, як комплементарну функцію помилок від кореня квадратного з відношення $\boldsymbol{m}\cdot\boldsymbol{m}$ («енергії сигналу») до дисперсії значень пікселів яскравості («енергія шуму») [5, 115].

У детекторі можливе виникнення двох типів помилок. Існує ймовірність того, що при аналізі контейнера детектор не виявить наявного в ньому прихованого повідомлення (β-імовірність пропускання цілі або так звана *помилка 2-го роду*), а також імовірність помилкового виявлення прихованого повідомлення в пустому контейнері (α-імовірність помилкової тривоги або так звана *помилка 1-го роду*). Зниження однієї ймовірності призводить до зростання іншої. На практиці для пасивного атакуючого намагаються максимізувати помилку 2-го роду. Ідеальна ж стеганосистема повинна забезпечувати помилку 2-го роду $\beta = 1$. Нижче буде показано, що всі абсолютно надійні стеганосистеми мають цю властивість (за умови, що атакуючий робить помилку 1-го роду з імовірністю $\alpha = 0$).

Для ρ-надійних стеганографічних систем імовірності α і β пов'язані між собою відповідно до поданої нижче теореми [15].

Теорема 3.2

Нехай Σ — стеганографічна система, яка є ρ-надійною проти пасивних атак. Тоді ймовірність β того, що порушник не виявить приховане повідомлення, і ймовірність α того, що він помилково виявить неіснуюче приховане повідомлення, задовольняють співвідношення $d(\alpha, \beta) \le \rho$, де $d(\alpha, \beta)$ — відносна двійкова ентропія, що визначається як

$$d(\alpha,\beta) = \alpha \cdot log_2 \frac{\alpha}{1-\beta} + (1-\alpha) \cdot log_2 \frac{1-\alpha}{\beta}. \qquad (3.22)$$

Зокрема, якщо $\alpha = 0$, то $\beta \ge 2^{-\rho}$.

Доведення.

У випадку, коли контейнери *не містять* прихованого повідомлення, їхній розподіл імовірностей відповідає P_C. Розглянемо випадкову величину $f(\mathbf{C})$ і обчислимо її імовірність π_C. Якщо $f(\mathbf{C}) = 1$ (контейнер приймається заповненим), порушник робить помилку 1-го роду. Таким чином, $\pi_C(1) = \alpha$ і $\pi_C(0) = 1 - \alpha$.

Якщо контейнер *містить* приховане повідомлення, то розподіл імовірності відповідає P_S. Обчислимо імовірність π_S для $f(\mathbf{S})$. У випадку, коли $f(\mathbf{S}) = 0$ (заповнений контейнер приймається за порожній), порушник робить помилку 2-го роду. Отже, $\pi_S(0) = \beta$ і $\pi_S(1) = 1 - \beta$. Відносна ентропія $D\left(\pi_C \| \pi_S\right)$, відповідно до (3.20) може бути виражена в такий спосіб:

$$D\left(\pi_C \| \pi_S\right) = \sum_{q \in \{0;1\}} \pi_C(q) \cdot log_2 \frac{\pi_C(q)}{\pi_S(q)} =$$

$$= \alpha \cdot log_2 \frac{\alpha}{1-\beta} + (1-\alpha) \cdot log_2 \frac{1-\alpha}{\beta} = d(\alpha,\beta).$$

Відзначимо таку властивість функції відносної ентропії: детермінована обробка не може збільшити відносну ентропію між двома розподілами. Нехай Q_0 і Q_1 — дві випадкові величини, визначені на множині спостережень **Q** з відповідними розподілами імовірностей P_{Q_0} і P_{Q_1}, а f — довільна функція відображення, що перетворює множину спостережень **Q** на множину спостережень **T** ($f: \mathbf{Q} \to \mathbf{T}$). Тоді є справедливим наступний вираз:

$$D\left(P_{T_0} \middle\| P_{T_1}\right) \le D\left(P_{Q_0} \middle\| P_{Q_1}\right),$$

де через T_0 і T_1 ($T_0, T_1 \in \mathbf{T}$) відповідно позначені випадкові величини $f(Q_0)$ і $f(Q_1)$. Отже, $d(\alpha,\beta) = D\left(\pi_C \middle\| \pi_S\right) \le D\left(P_C \middle\| P_S\right) \le \rho$. Враховуючи, що $\lim\limits_{\alpha \to 0} log_2[\alpha/(1-\beta)] = 0$, отримуємо: $d(0,\beta) = log_2(1/\beta)$. Отже, якщо $\alpha = 0$, то $\beta \ge 2^{-\rho}$. ***Теорему доведено.***

Таким чином, для ρ-надійної стеганосистеми з $\alpha = 0$, необхідно забезпечити, щоб $\rho \to 0$. При цьому імовірність $\beta \to 1$, що еквівалентне неможливості виявлення порушником прихованого у контейнері повідомлення.

3.7. Активні і зловмисні атаки

Навіть за умови неможливості виділення і читання прихованого повідомлення, факт наявності останнього можна відносно легко виявити. Ще легшою є операція знищення даного повідомлення. Наприклад, якщо повідомлення приховане у файлі формату *BMP* методом заміни палітри, то вплив на цей файл випадковою зміною кольорів у палітрі зробить повідомлення таким, що не видобувається, іншими словами — знищить його.

Під час проектування та дослідження стеганографічних систем особлива увага має бути приділена вивченню впливу на них активних і зловмисних атак [3]. *Активні атаки* здатні змінити контейнер під час зв'язку: порушник може перехопити надісланий контейнер-результат, змінити його (шляхом застосування певної обробки) і відправити далі приймальній стороні. У даному випадку передбачається, що при активній атаці замінити контейнер і його семантику повністю є *неможливим, а можна лише провести незначні модифікації таким*

чином, щоб оригінал і змінений контейнер залишалися візуально і семантично подібними. *Зловмисними атаками* є такі, за яких існує можливість підміни повідомлення іншим, тобто організовується фальсифікований стеганографічний обмін під ім'ям одного з партнерів зв'язку.

Стеганографічні системи є надзвичайно чутливими до модифікацій контейнера (наприклад, для зображення — це може бути згладжування і масштабування, для звуку — фільтрація тощо). Так, просте стискання із втратами може призвести до повної втрати інформації стеганограми, оскільки при цьому вилучаються незначущі компоненти сигналу і, тим самим, знищується секретна інформація, яка була в них прихована. Під час активних атак, коли видобути приховану інформацію або довести її існування немає можливості, її можна знищити простим додаванням до контейнера випадкових завад. У разі цифрових зображень атакуючий може застосувати поширені методи обробки зображень, у тому числі — просто змінити його формат.

У сучасних комп'ютерних системах реалізуються стеганографічні перетворення з високою надлишковістю, які є стійкими до трансформації контейнера (обертання, масштабування, друкування з наступним скануванням тощо). Тому однією з важливих вимог до прикладної стеганосистеми є забезпечення стійкості до випадкових або навмисних атак.

3.8. Стійкість стеганосистеми до активних атак

Виходячи з розглянутих вище особливостей атак на стеганосистеми, можна зробити висновок, що протидія статистичному стеганоаналізу має полягати в побудові математичних моделей сигналів-контейнерів, пошуку на їх основі «дозволених» для модифікації областей і вбудовуванню до них таких повідомлень, чия статистика не відрізняється від статистики контейнера. Така нерозрізненість і визначатиме стійкість стеганосистеми.

Як і для криптографічних систем захисту інформації, безпека стеганосистем описується й оцінюється їх *стійкістю* (стеганографічною стійкістю або скорочено *стеганостійкістю*). Але визначення стійкості і зламу для даних систем є різними. У криптографії система захисту інформації є стійкою, якщо володіючи перехопленою криптограмою, порушник не здатний видобути інформацію, що в ній міститься.

Неформально можна визначити, що стеганосистема є *стійкою*, якщо порушник, спостерігаючи за інформаційний обміном між від-

правником і одержувачем, не здатен виявити, що під прикриттям контейнерів передаються приховані у них повідомлення, і, тим паче, — дізнатися про зміст останніх. У більш широкому сенсі, під стійкістю стеганосистем розуміється їх здатність приховувати від кваліфікованого порушника факт організації стеганоканалу, здатність протистояти спробам порушника зруйнувати, спотворити, видалити приховано передавані повідомлення, а також спроможність підтвердити чи спростувати автентичність приховано передаваної інформації.

Стеганографічна система є стійкою до *активних атак*, якщо прихована за її допомогою інформація не може бути змінена без настільки значних змін контейнера, внаслідок яких останній втратить свою функціональність [3].

Означення 3.2

Нехай Σ — стеганографічна система і φ — клас відображень $\mathbf{C} \to \mathbf{S}$. Тоді система Σ буде φ-стійкою, якщо у випадку стеганосистем із секретним ключем для всіх $f \in \varphi$ є справедливим

$$D\{f[E(c, m, k), k]\} = D[E(c, m, k), k] = m, \qquad (3.23)$$

а у випадку безключових стеганосистем, незалежно від вибору $m \in \mathbf{M}$, $c \in \mathbf{C}$ і $k \in \mathbf{K}$:

$$D\{f[E(c, m)]\} = D[E(c, m)] = m. \qquad (3.24)$$

Очевидно, що існує і зворотній взаємозв'язок між *надійністю* стеганосистеми та її *стійкістю:* чим більш стійкою до модифікацій контейнера буде стеганосистема, тим вона буде менш надійною, оскільки стійкість може бути досягнута завадостійким кодуванням, що може призвести до суттєвого спотворення контейнера і, можливо, до зміни розподілу імовірності P_S.

Багато стеганосистем є стійкими лише до певного класу відображень (стискання із втратами, геометричні перетворення, фільтрація, переквантування відліків, додавання білого шуму, перетворення «цифра → аналог → цифра» тощо). Ідеальна стеганосистема повинна бути стійкою до всіх відображень типу «збереження λ-подібності», тобто відображенням f: $\mathbf{C} \to \mathbf{S}$ із властивістю $sim[c, f(c)] \to \lambda$ і $\lambda \approx 1$. Але такі системи є складними в проектуванні і, через необхідність додаткового застосування завадостійкого кодування, мають занадто низьку пропускну здатність. З іншого боку, система є «λ-слабкою», якщо для кожного контейнера існує таке відображення «збереження λ-подібності», *що прихована інформація буде невідновлюваною з точки зору співвідношень (3.23) або (3.24).*

У загальному випадку, існує два підходи до створення стійких стеганосистем [3]:

- передбачаючи можливі атаки на стеганограми з боку порушників, стеганографічне перетворення відразу проектується стійким до знищення прихованих даних певним класом модифікацій;
- реалізуються перетворення, які мають властивість зворотності до можливих модифікацій з метою відновлення початкового вигляду стеганограми. Зокрема, у [14] запропоновано метод «афінного кодування» для протидії афінним перетворенням зображення. При цьому передбачається оцінка параметрів перетворень, вимірювання змін форми, розмірів і напрямків деяких кодованих образів.

Стійкі алгоритми повинні приховувати дані в найбільш суттєвих фрагментах контейнера, оскільки інформація, яка кодується в шумоподібних компонентах, з легкістю може бути вилучена. Наприклад, відомо [57], що стеганографічні перетворення, що працюють з частотною областю контейнера, у своїй переважній більшості є стійкішими до модифікацій, порівняно з алгоритмами, які працюють у просторовій (для зображення) або часовій (для звуку) областях. Використовуючи ці властивості, можна створити стійкі стеганосистеми, які, наприклад, зберігатимуть приховану інформацію в коефіцієнтах дискретного косинусного перетворення (ДКП) зображення.

3.9. Свідомо відкритий стеганоканал

Вище було розглянуто моделі стеганосистем, при яких використовується секретний ключ (*private-key*), що розділяється поміж визначеними учасниками стеганографічного обміну. Для *пасивного* порушника подібні моделі є доволі обмеженими, і тим не менш, за певних обставин він все одно здатний виявити факт передавання прихованого повідомлення і навіть дізнатися про його зміст. Інша справа, коли порушник є *активним* або *зловмисним*. Тоді він не лише може вносити до стеганоканалу завади, але й навіть повністю імітувати відправника. Оскільки у більшості випадків апріорна інформація про відправника в одержувача є відсутньою, він стає неспроможним викрити фальсифікацію. Тому, здійснення прихованого передавання даних за наявності активного порушника є набагато складнішою проблемою, порівняно з фактом присутності пасивного порушника.

У роботах [5, 12] представлено протокол, що дозволяє вирішити цю задачу. Він заснований на введенні до розгляду каналу з винятково малою пропускною здатністю — свідомо відкритого каналу (*supra-*

liminal channel). Такий канал утворюється за рахунок вбудовування прихованих даних у найважливіші ознаки контейнера, спотворення яких призводитиме до повної деградації останнього. Іншими словами, інформація до контейнера вбудовується таким чином, що її видно, але неможливо змінити без істотних змін характерних властивостей останнього.

Очевидною є і сфера використання зазначеного протоколу — концепція відкритого стеганоканалу використовується переважно для вбудовування ЦВЗ. Розглянемо принцип його використання.

Припустимо, при активній атаці порушникові вдається внести тільки незначні зміни до перехопленого контейнера. Тоді з великою часткою імовірності деяка специфічна (основоположна) для цього контейнера інформація збережеться, оскільки її не можливо видалити без істотної зміни його семантики: у багатьох випадках порушник не може вносити завади в стеганоканал, які були б значними настільки, щоб стеганограма була повністю спотворена. Не може не через причини технічного характеру, а з юридичних чи інших міркувань. Тому, якщо секретне повідомлення вбудувати в істотні фрагменти контейнера, то його можна передавати між абонентами з високим ступенем цілісності навіть за наявності активної цьому протидії.

Вбудовування інформації до найбільш вагомих елементів контейнера є основним принципом застосування ЦВЗ. Характерна риса свідомо створюваного відкритого каналу полягає у тому, що для вбудовування і видобування даних секретний ключ стає непотрібним. Місце розміщення прихованих бітів є загальновідомим, але їх неможливо видалити без помітного (або відчутного) руйнування контейнера. Крім того, ЦВЗ може й не переносити у собі жодної осмисленої інформації — наприклад, бути певною функцією самого зображення. У випадку ж свідомого стеганоканалу, навпаки, контейнер може бути функцією прихованого короткого повідомлення.

Припустимо, що кожному контейнерові відповідає певний шаблон, що формально описує усі характерні особливості контейнера [3]. Нехай $\mathbf{S}$ — множина всіх шаблонів, а $\boldsymbol{f}: \mathbf{S} \to \{0;1\}^N$ — функція шаблонів. Для того, щоб передати бітовий рядок даних $\boldsymbol{m} \in \{0;1\}^N$, передавальна сторона обирає з множини $\mathbf{S}$ деякий шаблон $\boldsymbol{s} \in \boldsymbol{f}^{-1}(\boldsymbol{m})$ і надсилає відкритим каналом контейнер, якому цей шаблон відповідає. Пасивний порушник може підозрювати про існування прихованого обміну в шумовому компоненті контейнера і з метою руйнування секретного повідомлення може дещо змінити стеганограму. Однак при цьому змінити сам шаблон контейнера він не в змозі. У свою чергу, приймальна сторона може відновити шаблон $\boldsymbol{s}$ із прийнятої стеганограми (мож-

ливо, навіть модифікованої пасивним порушником) і видобути дані m за допомогою функції f.

Використовувати відкритий стеганоканал для пересилання бітового рядка m з явним змістом є недоцільним, оскільки пасивний порушник за допомогою відкритої функції f може легко відновити вбудовану інформацію. Однак, якщо повідомлення m є випадковим секретним ключем або ж виглядає як випадковий шифртекст (тобто попередньо було криптографічно захищене), то у порушника не буде ані підстав для підозри, ані доказів (якщо він не здатний зламати криптосистему), що передана інформація є чимось більш істотним, ніж випадкова бітова послідовність.

Для практичної реалізації даного протоколу необхідно забезпечити виконання низки умов. Очевидно, що основні труднощі полягають у формуванні контейнера, тоді як робота по видобуванню даних може бути легко автоматизованою. Для практичного ж застосування свідомо відкритого стеганоканалу мають бути автоматизовані обидві операції. По-перше, повинна існувати можливість створення для будь-якого шаблона такого контейнера, невеликі зміни якого при активних атаках не призводитимуть до зміни прихованих даних. При цьому у пасивного порушника не повинно існувати можливості шляхом маніпуляцій зі стеганограмою змінити шаблон s і привести його до такого вигляду s', що $f(s) \neq f(s')$. По-друге, повинна існувати можливість формування шаблону для кожного одержаного контейнера. Крім того, функція f має бути загальнодоступною, а f і f^{-1} — обчислюваними. Видобування повідомлення з пустого контейнера повинне повертати випадковий рядок даних. Отже, єдина відмінність між заповненим і порожнім контейнерами полягає у тому, що рядок $f(s)$ має певне осмислене значення.

Описана схема може бути також застосована для прихованого обміну ключами за наступним протоколом [12].

Передавальна сторона генерує пару відкритого (E) і секретного (D) ключів; обчислює представницький опис (шаблон) контейнера, що відповідає ключу E: $s_E \in f^{-1}(E)$; генерує контейнер, який відповідає s_E, і надсилає його приймальній стороні.

Приймальна сторона видобуває з прийнятого контейнера відкритий ключ E передавальної сторони: $E \in f(s_E)$; генерує свій секретний ключ K; шифрує його за допомогою відкритого ключа E; знаходить відповідний отриманій послідовності K_E опис (шаблон) контейнера: $s_{KE} \in f^{-1}(K_E)$; генерує контейнер, який відповідає s_{KE}, і надсилає його передавальній стороні.

Передавальна сторона відновлює зашифрований секретний ключ: $K_E \in f(s_{KE})$ і розшифровує його, використовуючи свій секретний ключ D.

Тепер сторони можуть обмінюватися повідомленнями, вбудовуваними до контейнера з використанням секретного ключа ***K***. Порушник в результаті перехоплення даних з каналу може отримати відкритий ключ передавальної сторони і зашифрований цим ключем секретний ключ приймальної сторони. Значення останнього без знання секретного ключа передавальної сторони залишатиметься для нього невідомим.

Слід зазначити, що головна проблема схеми свідомо відкритого стеганографічного каналу полягає в труднощах ефективної реалізації функції ***f***. Крім того, такий канал не підходить для прихованого передавання повідомлень великого об'єму, оскільки він має низьку пропускну здатність і є відкритим для порушника.

Луїс фон Ан *(L. von Ahn)* та Ніколас Гоппер *(N.J. Hopper)* [58] формалізували стеганографію з відкритим ключем у випадку пасивного порушника, а також створили обмежену модель за наявності порушника, який здійснює активну атаку на стеганоканал. На їхню думку, стійкість проти атак конкретних (заздалегідь визначених) порушників необхідно забезпечувати у тих випадках, коли одержувач має бути впевненим в автентичності відправника. Однак такий підхід, на думку авторів [59], є обмеженням моделі порівняно із принципами, закладеними у процес обміну з відкритим ключем. У своїй роботі вони пропонують комплексну теоретичну модель протоколу стеганографічного обміну з відкритим ключем у випадку активних атак, причому особи, які беруть у ньому участь, апріорі не потребують розділення поміж собою секретної інформації, а порушник може впливати на канал і встановлювати так звану *адаптивну до контейнерів атаку*. Такий вид атаки бачиться найбільш узагальнюючим проти стеганосистем, протоколи яких побудовані з використанням відкритості ключа. Це дозволяє порушнику надсилати приймальній стороні довільну послідовність адаптивно обраних вбудованих у контейнер повідомлень і вивчати інтерпретацію кожного з повідомлень; тобто, якщо одержувач розглядає повідомлення як пустий контейнер або ж як стеганограму і видобуває приховане повідомлення в останньому випадку. Описана у [59] модель побудована на припущенні, що стеганосистема з відкритим ключем за своєю сутністю є криптографічною системою з відкритим ключем з додатковою необхідною умовою, що *результат її роботи (стеганограма) повинна відповідати розподілові використаного при цьому контейнера.*

3.10. Підсумки розділу

У даному розділі шляхом дослідження відомих публікацій здійснено системне викладення питань надійності і стійкості довільної стеганографічної системи по відношенню до різноманітних видів здійснюваних на неї атак. Останні було поділено у відповідності до класифікації типів порушників на пасивних, активних і зловмисних.

Проведення аналогії між стегано- і криптоаналізом дозволило виділити як спільні, так і характерні лише для стеганосистем види атак (атака на основі відомого порожнього контейнера, атака на основі обраного порожнього контейнера, атака на основі відомої математичної моделі контейнера або ж його фрагмента).

Також було здійснено й огляд низки публікацій, присвячених розглядові показників, використовуваних з метою оцінювання спотворень, що вносяться стеганоперетвореннями до структури контейнера.

Розділ 4. ПРОПУСКНА ЗДАТНІСТЬ СТЕГАНОГРАФІЧНИХ КАНАЛІВ

4.1. Поняття прихованої пропускної здатності

Для розроблюваних або вже досліджуваних стеганографічних систем важливо визначити, наскільки ж великою може бути пропускна здатність створюваних при цьому каналів передавання прихованих даних (КППД), і як вона залежатиме від інших характеристик стеганосистем та умов їхнього використання. Під пропускною здатністю КППД або просто *прихованою пропускною здатністю* (ППЗ) розуміють максимальну кількість інформації, що може бути вбудована до одного елемента (наприклад, пікселя чи аудіовідліку) контейнера. Обов'язковою умовою при цьому є безпомилковість передавання прихованих даних одержувачеві, а також їх захищеність від таких атак порушника, як спроби виявлення факту наявності каналу прихованого зв'язку, одержання змісту прихованих повідомлень, навмисне введення сфальсифікованих даних або ж руйнування вбудованої до контейнера інформації [5].

Канал прихованого зв'язку (КПЗ) утворюється всередині каналу відкритого зв'язку (КВЗ), пропускну здатність якого у своїх роботах з теорії інформації визначив ще Клод Шеннон *(C. E. Shannon)* [45, 60, 65, 70, 115]. Пропускна здатність КВЗ визначається як кількість інформації, що її потенційно можна передати без помилок за одне використання каналу. При цьому не висувається жодних вимог до захищеності від атак організованого порушника. Тому цілком логічним є припущення, що прихована пропускна здатність КПЗ, в якому за одне використання каналу передається один елемент контейнера із вкладеною у нього прихованою інформацією, у жодному випадку не може виявитися більшою за пропускну здатність КВЗ.

На сьогодні існують різні, іноді діаметрально протилежні підходи до визначення кількості інформації, яка підлягає захистові від різноманітних атак порушника саме за допомогою стеганографічних методів. Дані розходження, як зазначається у [5], зумовлені відмінністю у цілях захисту інформації, видами порушника, його можливостями та реалізованими ним атаками на стеганосистеми, видом використовуваних контейнерів і прихованих повідомлень та багатьма іншими факторами. У тій самій роботі наводиться оцінювання величини ППЗ стеганоканалу методами теорії інформації для різних стеганосистем. Теоретико-інформаційні методи дозволяють одержати суворі оцінки

кількості прихованої інформації, які цілком правомірно можуть бути використані як теоретично досяжні граничні швидкості передавання прихованої інформації для стеганографічних систем, не зважаючи на принципи, закладені до основи їхньої побудови.

Розглянемо два основних підходи до оцінки пропускної здатності КППД. Перший з них, розвинутий у роботах [61, 62], орієнтований на стеганографічні системи, в яких повідомлення, що підлягають прихованню, повинні бути безпомилково передані в умовах активної протидії порушника. Даний підхід описує сценарій стеганографічного вбудовування так званих безнадлишкових повідомлень у даних контейнера, і, що найголовніше, дозволяє враховувати той факт, що крім спотворювань структури контейнера при вбудовуванні до нього даних повідомлення, можливі і його навмисні спотворювання з боку порушника. Крім того, існує ще й певна імовірність спотворень випадкового характеру, викликаних ненавмисними завадами у каналі зв'язку.

Порушник крім пасивних дій аналізу може використати й активні дії (атакуючий порушник). Метою атакуючого порушника є руйнування прихованої інформації. Така постановка завдання інформаційного приховання є характерною, наприклад, для систем з ЦВЗ.

У [5] сформульовано задачу інформаційного приховання як завдання безпомилкового передавання прихованої інформації під впливом випадкових і навмисних завад, а також визначено максимальну швидкість безпомилкового передавання за різних стратегій дій відправника й атакуючого. Пропонований підхід визначає *теоретично досяжну* швидкість достовірного передавання прихованих повідомлень, хоча в явному виді й не оцінює захищеність останніх від виявлення факту їх існування. Разом з тим, для низки стеганосистем не потрібно приховувати факт використання стеганографічного захисту: власник авторських або майнових прав на мультимедійний, захищений ЦВЗ, контейнер, як правило, відкрито повідомляє про застосування даної технології захисту. У розглянутому підході досліджуються умови, за яких прихована інформація гарантовано передається в умовах довільних спроб порушника щодо її руйнування.

Знання параметрів стеганосистеми і можливих стратегій дій передавальної сторони не повинне дозволити порушникові оптимізувати руйнуючий вплив та оцінити його ефективність. Особливістю таких стеганосистем є, по-перше, те, що руйнуючий вплив відбувається лише у момент передавання прихованих даних і повинен здійснюватися у режимі реального часу. По-друге, існує апріорна непоінформованість законного одержувача щодо приховано передаваної йому інформації. По-третє, у переважній більшості випадків порушник не спроможний достовірно оцінити ефективність своїх дій.

Геть інша ситуація виникає при намаганні активного порушника зруйнувати ЦВЗ з метою привласнити собі контейнер (точніше — права на нього). Порушник може як завгодно довго здійснювати руйнуючий вплив, обираючи таку оптимальну стратегію, за якої, зруйнувавши ЦВЗ, він збереже достатню для себе якість контейнера. При цьому порушник заздалегідь знає про існування прихованої інформації, і, використовуючи загальновідомий детектор (див. рис. 3.3), здатний оцінити ефективність своїх атак на ЦВЗ.

Другий підхід, що пропонується, наприклад, авторами робіт [19, 44, 46], дає оцінки ППЗ безпосередньо в процесі вбудовування прихованих повідомлень у надлишкові дані контейнера. Такий підхід враховує, що контейнери формуються реальними надлишковими джерелами з істотною пам'яттю, такими як джерела зображень або аудіосигналів. У цьому випадку оцінки ППЗ залежать від характеристик замаскованості прихованого каналу. Такий підхід є орієнтованим на стеганосистеми, в яких реалізується приховане передавання апріорно невідомої одержувачеві інформації, причому в процесі спостереження за КВЗ пасивний порушник намагається виявити факт наявності у ньому КПЗ і, в разі встановлення цього факту, прагне розкрити зміст прихованого повідомлення у перехопленому контейнері.

Відома велика кількість робіт із синтезу стеганосистем, автори яких пропонують різні способи вбудовування даних у надлишкові за своєю природою контейнери [21, 24, 63, 64]. При цьому кількість інформації, вбудованість якої залишатиметься непомітною, оцінюється за допомогою додатково введених критеріїв рівня прихованості (див. розділ 3). Існуючі на сьогодні оцінки ППЗ таких стеганоканалів, однак, не враховують можливі випадкові чи навмисні спотворення контейнерів при їх передаванні каналом зв'язку.

4.2. Інформаційне приховання при активній протидії

У рамках першого підходу до оцінки ППЗ розглянемо загальне формулювання завдання інформаційного приховання у випадку активної протидії з боку порушника. Основні результати цього підходу були отримані в роботі [61] і застосовані у [5]. Далі наведемо деякі з них.

4.2.1. Формулювання завдання інформаційного приховання при активній протидії

Розглянемо представлену на рис. 4.1 узагальнену структурну схему стеганографічної системи передавання прихованих повідомлень.

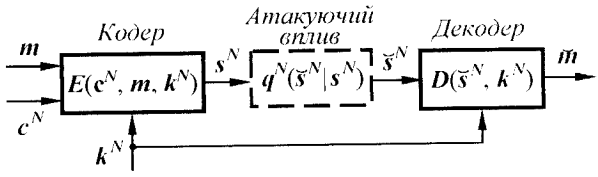


Рис. 4.1. Узагальнена структурна схема стеганосистеми при активній протидії порушника

У схемі на рис. 4.1 приховані повідомлення m рівномірно розподілені у множині повідомлень $\mathbf{M}$ і мають бути безпомилково передані декодерові. Передавальна сторона подає порожній контейнер c^N (що являє собою послідовність з N незалежно й ідентично розподілених відліків у відповідності до розподілу контейнера $p(c)$), секретний ключ k^N (кожен символ k_i якого незалежно і рівноймовірно розподілений за функцією $p(k)$), і повідомлення m на вхід кодера. Останній формує контейнер-результат s^N, що передається одержувачеві незахищеним каналом зв'язку.

Контейнер s^N перехоплюється й обробляється порушником з метою руйнування або видалення повідомлення m. Спотворений порушником контейнер позначимо як $\breve{s}^N$, а атакуючий вплив — апостеріорною функцією розподілу $q^N\left(\breve{s}^N \middle| s^N\right)$. Як окремий випадок, ця обробка включає формування спотвореного контейнера-результату у вигляді $\breve{s}^N = q^N\left(s^N\right)$, де q^N — детерміноване відображення.

Основне припущення наступне: порушник знає розподіли всіх змінних у стеганосистемі та, власне, опис цієї стеганосистеми, але не знає використовуваного секретного ключа (тобто принцип Керкгоффса для систем захисту інформації дотримано).

Нехай контейнер-оригінал c, контейнер-результат s і модифікований порушником контейнер-результат $\breve{s}$ належать спільній множині $\mathbf{C}$ ($c, s, \breve{s} \in \mathbf{C}$). Декодер одержувача визначає оцінку $\breve{m}$ первинного прихованого повідомлення m. Якщо $m \neq \breve{m}$, то атакуючому вдалося зруйнувати інформацію, що захищалася стеганографічною системою.

Формально визначимо внесені спотворювання в стратегіях передавальної сторони і порушника. Це завершує математичний опис стеганосистеми і дозволяє визначити швидкість безпомилкового передавання для схеми, представленої на рис. 4.1.

Означення 4.1

Функція спотворення, що вноситься відправником повідомлення, являє собою невід'ємну функцію d_1: $\mathbf{C} \times \mathbf{S} \rightarrow \mathfrak{R}_+$.

Означення 4.2

Функція спотворення, що вноситься атакуючою стороною, являє собою невід'ємну функцію d_2: $\mathbf{S} \times \bar{\mathbf{S}} \to \Re_+$.

Функція спотворення d_1 є обмеженою: $d_{1,max} = \max\limits_{(c,s) \in \mathbf{C} \times \mathbf{S}} d_1(c,s) < \infty$.

Крім того, дана міра спотворення є симетричною: $d_1(c, s) = d_1(s, c)$ для всіх $c, s \in \mathbf{C} = \mathbf{S}$. Виконання рівності $d_1(c, s) = 0$ означає збіг: $c = s$. Якщо $d_1(c, s) = 1$, то контейнер-результат не відповідає контейнеру-оригіналу.

Функції спотворення d_i, $i \in \{1; 2\}$ поширюються на спотворення символьних послідовностей з довжиною блоків N:

$$d_i^N(x^N, y^N) = N^{-1} \cdot \sum_{j=1}^{N} d_i(x_j, y_j).$$

Назвемо спотворення контейнера c, викликане вбудовуванням у нього прихованого повідомлення m, *спотворенням, викликаним кодуванням*, а спотворення, викликане атакуючими діями порушника, — *спотворенням, викликаним атакуючим впливом*.

Означення 4.3

Стеганосистема з довжиною блока N, що призводить до викликаного кодуванням спотворення, рівень якого не перевищує A_1, є сукупністю множин прихованих повідомлень $\mathbf{M}$, контейнерів $\mathbf{C}$, стеганограм $\mathbf{S} \sim \bar{\mathbf{S}} \sim \mathbf{C}$ і ключів $\mathbf{K}$, а також визначених для них функцій кодування E і декодування D. Причому E — це відображення контейнера c^N, повідомлення m і ключа k^N до контейнера-результату: $E: \mathbf{C} \times \mathbf{M} \times \mathbf{K} \to \mathbf{C}$, $s^N = E(c^N, m, k^N)$.

Це відображення є обмеженим величиною середнього спотворення A_1, викликаного кодуванням:

$$\sum_{c^N \in \mathbf{C}} \sum_{m \in \mathbf{M}} \sum_{k^N \in \mathbf{K}} |\mathbf{M}|^{-1} \cdot p(c^N, k^N) \cdot d_1^N\left[c^N, E(c^N, m, k^N)\right] \le A_1. \quad (4.1)$$

Відображення $D: \mathbf{C} \times \mathbf{K} \to \bar{\mathbf{M}}$ є декодувальним відображенням прийнятої стеганопослідовності $\bar{s}^N$ і ключа k^N у декодоване повідомлення $m = D(\bar{s}^N, k^N)$.

Отже, величина A_1 характеризує той ступінь спотвореності контейнера, що є максимально припустимим при вбудовуванні до нього прихованого повідомлення. Незважаючи на те, що дане визначення формально описує стеганосистеми блокового типу, на практиці воно може

бути поширене і на стеганосистеми потокового типу, в яких вікно обробки описується ковзним блоком довжиною N. У цьому випадку параметр N може бути названий довжиною кодового обмеження стеганосистеми (за аналогією з безперервними кодами [65, 115]).

У більшості випадків спотворення A_1 є малим, оскільки апріорно приймається, що результат вбудовування до контейнера повідомлення має бути невідчутним для стороннього пересічного спостерігача (у тому числі — й порушника). У стеганосистемах, в яких контейнер являє собою корисний для одержувача інформаційний сигнал і якість якого необхідно зберегти, величина A_1 обмежується. У системах ЦВЗ вимога мінімізації A_1 формулюється як вимога прозорості водяного знаку, що засвідчує про законну приналежність контейнера [5].

Крім того, визначення обмеження спотворення (4.1) містить усереднення по відношенню до розподілу $p(c^N, k^N)$ і по відношенню до рівномірного розподілу повідомлень. Такий вибір зроблено для зручності, оскільки це дозволяє використовувати класичні положення теорії Шеннона [60, 115].

Розподіл $p(c^N, k^N)$ та обрання відображення E визначають конкретний вигляд розподілу $p(s^N)$ множини формованих контейнерів-результатів.

Означення 4.4

Атакуючий вплив (без пам'яті), що призводить до спотворення A_2, описується умовною функцією розподілу $q^N(\bar{s}^N \mid s^N)$ з множини $\mathbf{S}$ до множини $\bar{\mathbf{S}}$, такою що

$$\sum_{s^N \in \mathbf{S}} \sum_{\bar{s}^N \in \bar{\mathbf{S}}} d_2^N(s^N, \bar{s}^N) \cdot q^N(\bar{s}^N \mid s^N) \cdot p(s^N) \le A_2. \qquad (4.2)$$

За визначенням, A_2 є максимальною величиною спотворення контейнера-результату, що було викликане навмисними діями порушника. Фізичний зміст обмеження величини A_2 полягає в наступному. У системах ЦВЗ порушник, намагаючись видалити водяний знак із завіреного останнім контейнера, змушений сам зменшувати величину A_2, щоб істотно не спотворити цінний для нього медіа-об'єкт. В інших стеганосистемах величина A_2 обмежується наявним в атакуючого енергетичним потенціалом встановлення завад, виникаючими при цьому завадами для інших каналів зв'язку у випадку використання спільного ресурсу та з низки інших причин.

Логічним є припущення, що для практичних стеганосистем зазвичай має виконуватися співвідношення $A_2 \ge A_1$. Відповідно до означення 4.4, атакуючий вплив описується й обмежується усередненими спотвореннями між множинами $\mathbf{S}$ і $\bar{\mathbf{S}}$. В інших випадках, якщо

порушник знає опис функції $\boldsymbol{E}$, то атакуючий вплив описується і обмежується усередненим спотворенням між множинами $\mathbf{C}$ і $\mathbf{S}$:

$$\sum_{\substack{c^N, m, \\ k^N, s^N}} d^N\left(c^N, s^N\right) \cdot q^N\left[\bar{s}^N \middle| E\left(c^N, m, k^N\right)\right] \cdot p\left(c^N, k^N\right) \le A_2. \qquad (4.3)$$

Відповідно до (4.3), визначення A_2 припускає, що порушникові відомі точні ймовірнісні характеристики контейнерів. Як буде показано далі, ця обставина суттєво ускладнює завдання забезпечення захищеності прихованої інформації, тому в стійких стеганосистемах використовуються різні методи приховання від порушника характеристик використовуваних контейнерів. Наприклад, такі методи включають використання для вбудовування підмножини контейнерів з імовірнісними характеристиками, що відрізняються від характеристик усієї множини відомих порушникові контейнерів або рандомізоване стискання сигналу контейнера перед вбудовуванням до нього прихованого повідомлення [15]. Тому обчислення спотворення A_2 у відповідності до означення 4.4 є більш універсальним, оскільки порушник завжди має можливість вивчати ймовірнісні характеристики перехоплюваних ним контейнерів.

Маючи опис стеганосистеми і атакуючого впливу $q^N(\bar{s}^N \mid s^N)$, можна описати змагання (гру) між передавальною і атакуючою сторонами.

Означення 4.5

Інформаційно-приховуюче змагання, що призводить до спотворювань (A_1, A_2), описується взаємодією використовуваної стеганосистеми, що спричиняє спотворення кодування A_1, і атакуючого впливу, що, у свою чергу, викликає спотворення A_2. Швидкість передавання прихованих повідомлень по стеганоканалу визначається як $R = N^{-1} \log |\mathbf{M}|$. При цьому швидкість передавання R виражається через середню кількість бітів прихованого повідомлення, які безпомилково передаються (переносяться) одним символом (пікселем, аудіовідліком) стеганопослідовності s^N.

Це визначення є співзвучним до «класичного» визначення швидкості передавання звичайних повідомлень каналом відкритого зв'язку, що виражається у середній кількості безпомилково переданих бітів за одне використання каналу [60, 65, 70, 115].

Імовірність руйнування прихованого повідомлення (середню імовірність помилки) в стеганопослідовності довжиною N визначають як

$$P^N_{\text{руйн}} = |\mathbf{M}|^{-1} \cdot \sum_{m \in \mathbf{M}} P\left[D\left(\bar{\mathbf{S}}, \mathbf{K}\right) \ne m \middle| \mathbf{M} = m\right], \qquad (4.4)$$

де приховані повідомлення m рівноймовірно обираються серед елементів множини **M**.

Імовірність $P^{\wedge}_{\text{руйн}}$ є середньою імовірністю того, що атакуючий успішно спотворить вбудоване до контейнера повідомлення. Усереднення відбувається по всій множині повідомлень **M**. Атакуючий зазнає успіху в інформаційному змаганні, якщо декодована одержувачем стеганограма не збігається з повідомленням, яке вбудовувалося до контейнера відправником, або ж коли декодер не здатен прийняти однозначне рішення.

Теоретично досяжну швидкість безпомилкового передавання прихованих повідомлень і, що те саме, приховану пропускну здатність при спотвореннях не більше за (A_1, A_2), пропонується визначити в такий спосіб.

Означення 4.6

Швидкість R безпомилкового передавання прихованих повідомлень є досяжною для спотворень, що не перевищують (A_1, A_2), якщо існує стеганосистема з довжиною блока N, що призводить до спотворень кодування не більше A_1 на швидкості $R_N > R$, така, що за будь-яких атак порушника, які призводять до спотворень не більше A_2, при $N \to \infty$ матиме наслідком те, що $P^{\wedge}_{\text{руйн}} \to 0$,

Означення 4.7

Прихована пропускна здатність $B(A_1, A_2)$ є супремумом (найменшою верхньою межею) усіх досяжних швидкостей безпомилкового передавання прихованих повідомлень при спотвореннях не більше за (A_1, A_2).

Таким чином, ППЗ є верхньою межею швидкості безпомилкового передавання прихованих даних, за якої спотворення контейнера A_1, викликані вбудовуванням у нього зазначених повідомлень, і A_2, викликані діями порушника по руйнуванню цих повідомлень, не перевищують заданих величин.

Як і ПЗ каналів передавання відкритих повідомлень, ППЗ стеганоканалів визначається в ідеалізованих умовах, при яких затримка кодування/декодування є нескінченною (тобто $N \to \infty$), статистика контейнерів, прихованих повідомлень, стеганограм і ключів є точно відомою, а складність побудови стеганосистеми — нічим не обмеженою.

Очевидно, що така пропускна здатність каналу прихованого зв'язку має сутність теоретичної межі, що вказує області, в яких існують і, відповідно, не існують стеганосистеми при заданих величинах спотворень. Відомо, що швидкості реальних систем передавання відкритих повідомлень можуть лише наближатися до ПЗ КВЗ [115], причому по мірі наближення до неї обчислювальна складність реалізації таких систем

зростає спочатку приблизно за лінійною, потім за квадратичною і далі за експонентною залежністю від довжини блока кодування N [60]. Ймовірно, що по мірі наближення швидкості передавання прихованих даних до величини ППЗ КПЗ аналогічні залежності зростання складності будуть справедливими і для стеганографічних систем зв'язку. Це припущення підтверджується наявним досвідом побудови стеганосистем [5]. Відомо також, що спроби збільшити швидкість передавання прихованих даних призводять до істотного ускладнення методів приховання інформації [63, 66].

4.2.2. Приховуюче перетворення

Для повного уявлення стеганосистеми та умов її функціонування наведемо формальний опис *приховуючого перетворення*, виконуваного під час вбудовування медіа-даних до контейнера, і *атакуючого впливу*, здійснюваного порушником для протидії прихованому передаванню. Для цього розглянемо допоміжну випадкову послідовність $\boldsymbol{u}$, визначену на множині $\mathbf{U}$.

Фізично послідовність $\boldsymbol{u}$ описує результат перетворення прихованого повідомлення $\boldsymbol{m}$ з метою його адаптації до вбудовування до контейнера заданого типу і формату. Слід зауважити, що в той час як у стеганосистемі пусті і заповнені контейнери, ключі являють собою послідовності однакової довжини N, довжина прихованих повідомлень, їхній алфавіт та ймовірнісний розподіл у переважній більшості випадків з відповідними характеристиками зазначених послідовностей не збігаються.

Визначимо допоміжну множину

$$\boldsymbol{O} = \{(c, k) \in \mathbf{C} \times \mathbf{K} : p(c, k) > 0\}.$$

Тоді потужність множини $\mathbf{U}$ повинна задовольняти умові:

$$|\mathbf{U}| \le |\mathbf{S}| \cdot |\mathbf{O}| + 1.$$

У загальному вигляді приховуюче перетворення, що використовується відправником для вбудовування прихованого повідомлення до контейнера, визначається наступним чином.

Означення 4.8

Приховуюче перетворення, що викликає спотворення кодування A_1, описується умовною функцією розподілу $\tilde{q}(s, u \mid c, k)$ відображення з множини $\mathbf{C} \times \mathbf{K}$ у множину $\mathbf{C} \times \mathbf{U}$, такою, що виконується умова

$$\sum_{c, s, k, u} d_1(c, s) \cdot \tilde{q}(s, u \mid c, k) \cdot p(c, k) \le A_1. \qquad (4.5)$$

Розширення приховуючого перетворення без пам'яті довжиною N описується наступною умовною функцією:

$$\tilde{q}^{N}\left(s^{N}, u^{N} \mid c^{N}, k^{N}\right) = \prod_{i=1}^{N} \tilde{q}\left(s_i, u_i \mid c_i, k_i\right). \qquad (4.6)$$

Для успішного приховання інформації від кваліфікованого порушника доцільно використовувати не одне приховуюче перетворення повідомлення, а деяку їх множину.

Означення 4.9

Узагальнене приховуюче перетворення, що спричиняє спотворення кодування не більше за величину A_1, складається з множини $\tilde{\psi}$ усіх приховуючих перетворень, які задовольняють умові (4.5).

Узагальнене приховуюче перетворення описує всі можливі варіанти дій відправника при вбудовуванні повідомлень m у контейнер таким чином, щоб величина спотворення кодування не перевищувала припустиму A_1. Слід зазначити, що в стеганографії важливо, щоб у приховуючого інформацію існувала множина можливих варіантів, серед яких він рівноймовірно й непередбачувано для порушника обирає якийсь конкретний варіант приховання повідомлення, що потребує захисту.

Для аналізу стеганосистеми функцію $\tilde{q}$ зручно записати у формі добутку функцій розподілу:

$$\tilde{q}(s, u \mid c, k) = p(s \mid c, u, k) \cdot p(u \mid c, k), \qquad (4.7)$$

де $p(s \mid c, u, k)$ відноситься до «основного» приховуючого перетворення, а $p(u \mid c, k)$ — до «допоміжного».

4.2.3. Атакуючий вплив

Розглянемо формальний опис дій порушника при перетворенні перехопленого контейнера s на спотворений контейнер $\bar{s}$ з метою руйнування прихованої інформації, що у ньому міститься.

Означення 4.10

Атакуючий вплив, що викликає спотворення A_2, описується апостеріорною функцією розподілу $q\left(\bar{s} \mid s\right)$ відображення з множини $\mathbf{S}$ у множину $\bar{\mathbf{S}}$, такою, що виконується умова

$$\sum_{s,\,\bar{s}} d_2(s, \bar{s}) \cdot q(\bar{s} \mid s) \cdot p(s) \le A_2 . \qquad (4.8)$$

Розширення атакуючого впливу без пам'яті довжиною N описується умовною функцією виду

$$q^{N}\left(\bar{s}^{N} \middle| s^{N}\right) = \prod_{i=1}^{N} q\left(\bar{s}_i \middle| s_i\right). \tag{4.9}$$

Означення 4.11

Узагальнений атакуючий вплив, що викликає спотворення не більше за величину A_2, складається з множини ψ всіх атакуючих впливів, які задовольняють умові (4.8).

Аналогічно набору варіантів дій передавальної сторони, в атакуючого також є свій набір атакуючих впливів (множина ψ). Порушник, перехопивши контейнер-результат, намагається обрати такий атакуючий вплив з множини ψ, який би максимізував імовірність руйнування прихованої в ній інформації.

4.3. Прихована пропускна здатність каналу при активній протидії порушника

4.3.1. Основна теорема інформаційного приховання при активній протидії порушника

Дослідимо ППЗ у випадку активної протидії порушника, який прагне зруйнувати передавану приховано інформацію. Інформаційно-приховуюче змагання між передавальною і атакуючою сторонами зручно описати методами *теорії ігор* [67, 68, 97].

У теорії ігор зазвичай мова йде про ігри саме двох сторін (осіб); при цьому вважається, що ігри є обмеженими, тобто кожен з гравців може робити лише певну кількість кроків, а сама гра завершується після обмеженої кількості кроків. Звідси випливає обмеженість кількості стратегій обох гравців. Під поняттям *стратегії* розуміється така система правил, за допомогою якої задається дія (або дії) гравця у певній ситуації. Метою теорії ігор є знаходження кращої стратегії окремих гравців.

Ціна гри дорівнює величині ППЗ, для максимізації якої відправник медіа-даних оптимально будує приховуюче перетворення. Для мінімізації ППЗ атакуючий синтезує оптимальний атакуючий вплив. Величина ППЗ може бути отримана послідовним з'єднанням приховуючого перетворення і атакуючого впливу. Для можливості оцінки величини ППЗ для стеганосистеми з двійковим алфавітом, дослідимо теоретико-ігрові аспекти проблеми приховання інформації стеганосистемами.

Розглянемо *основну теорему інформаційного приховання при активній протидії порушника* [61]. Для будь-яких стеганосистем довільної складності і будь-яких атак без пам'яті дана теорема обмежує зверху швидкість безпомилкового передавання для приховуючого інформацію за умови, що атакуючий знає опис приховуючого перетворення, а одержувач знає описи як приховуючого перетворення, так і атакуючого впливу. Дана умова насправді не є важкоздійсненною, як це може здатися на перший погляд. Навіть якщо стратегії дій відправника інформації та атакуючого є невідомими, але стаціонарними, то можна стверджувати, що як атакуючий, так і одержувач потенційно здатні їх визначити, обробивши достатньо великий обсяг статистичного матеріалу. Це припущення є цілком реалістичним, хоча й не завжди може бути досягнуте на практиці з огляду на високу обчислювальну складність [5].

Попередньо розглянемо два твердження, що встановлюють області існування стеганосистем, потенційно здатних безпомилково передавати приховану інформацію при заданому атакуючому впливі [5].

Позначимо через $H(x)$ ентропію змінної x, через $I(x; y)$ — апріорну повну кількість інформації між x та y, а через $I(x; y \mid z)$ — апостеріорну повну кількість інформації між x та y, зумовлених z [68].

Твердження 4.1

Зафіксуємо атакуючий вплив $q(\vec{s} \mid s)$ й оберемо приховуюче перетворення $\tilde{q}(s, u \mid c, k)$, яке максимізує кількість інформації виду

$$J(\tilde{q}, q) = I(u; s \mid k) - I(u; c \mid k) \qquad (4.10)$$

над $\tilde{\psi}$. Для будь-якого як завгодно малого значення $\xi > 0$ і достатньо великого значення N існує така стеганосистема з довжиною блока N, що забезпечує ймовірність руйнування прихованих повідомлень $P_{\text{руйн}}^{\setminus} < \xi$ для множини прихованих повідомлень потужністю

$$|\mathbf{M}| < 2^{N \cdot \left[I(u; s \mid k) - I(u; c \mid k) - \xi \right]}.$$

Твердження 4.2

Нехай стеганосистема з довжиною блока N здатна безпомилково передавати приховані повідомлення зі швидкістю $R = N^{-1} \cdot \log |\mathbf{M}|$ біт/елемент контейнера при атакуючому впливі $q(\vec{s} \mid s)$. Якщо для будь-якого $\xi > 0$ стеганосистема забезпечує ймовірність руйнування прихованих повідомлень $P_{\text{руйн.}}^{\setminus} < \xi$ при $N \to \infty$, то існує такий кінцевий

алфавіт **U** і таке приховуюче перетворення $\tilde{q}(s, u \mid c, k)$, що виконується нерівність

$$R \le \dot{I}(u; \bar{s} \mid k) - \dot{I}(u; c \mid k).$$

Звідси:

Теорема 4.1

Якщо атакуючому відомий опис узагальненого приховуючого перетворення $\tilde{\psi}$, а одержувачеві крім цього відомий ще й опис узагальненого атакуючого впливу ψ, то для будь-якого інформаційно-приховуючого змагання, що призводить до спотворювань не більше за (A_1, A_2), швидкість передавання прихованих повідомлень R є досяжною тоді і тільки тоді, коли виконується умова $R < \underline{B}$.

Величина $\underline{B}$ визначається як

$$\underline{B} = \max_{\tilde{q}(s, u \mid c, k) \in \tilde{\psi}} \min_{q(s \mid s) \in \psi} J(\tilde{q}, q), \qquad (4.11)$$

де u — випадкова змінна над довільним кінцевим алфавітом **U**. Змінні $(u, c, k) \to s \to \bar{s}$ утворюють марківський ланцюг, що являє собою окрему форму наступного марківського ланцюга: $u \to (c, s) \to \bar{s}$, характеристики якого розглядаються у [69]. Кількість інформації $J(\tilde{q}, q)$ визначається виразом (4.10).

Таким чином, теорема 4.3 визначає величину *нижньої межі* ППЗ в умовах, коли всі учасники інформаційного змагання знають стратегії дій один одного. Треба зауважити, що в даній теоремі визначається величина ППЗ стеганоканалу, про існування якого атакуючому відомо. Дана ППЗ дорівнює середній кількості бітів інформації на один елемент контейнера, яку порушник не може зруйнувати, обираючи будь-яку стратегію протидії з наявної множини ψ при спотворенні контейнера не більше величини A_2.

Доведення цієї теореми зводиться до наступного [5].

Зафіксуємо атакуючий вплив $q \in \psi$. У твердженні 4.1 говориться, що всі швидкості безпомилкової передачі прихованих повідомлень, менші за $\max_{\tilde{q} \in \tilde{\psi}} J(\tilde{q}, q)$, є досяжними. Твердження 4.2 містить зворотний результат, тобто достовірне передавання вище цієї швидкості є неможливим. Оскільки атакуючий обізнаний з розподілом $\tilde{q}$, то він здатен обрати такий розподіл q, який мінімізуватиме швидкість передачі. ***Теорему доведено.***

Далі показано, що у важливому спеціальному випадку, коли $k = c$ (тобто секретним ключем стеганосистеми є опис використовуваного контейнера, а сам контейнер відомий одержувачеві), при обмеженні кодера стеганосистеми видом, представленим на рис. 4.1 втрати в оптимальності відсутні.

Наслідок

У випадку $k = c$ вибір значення змінної u є оптимальним тоді і тільки тоді, коли стеганосистема s може бути записана у формі $s = E(c, u)$, де відображення $E(c, \bullet)$ є зворотним для всіх значень c. Зокрема, вибір $u = s$ є оптимальним. ППЗ у цьому випадку визначається наступним чином:

$$B = \max_{p(s|c)} \min_{q(s|s)} \dot{I}(s; s \mid c) = \min_{q(s|s)} \min_{p(s|c)} I(s; s \mid c). \qquad (4.12)$$

Це випливає з того, що коли $k = c$, вираз (4.10) може бути записаний у вигляді

$$J(\tilde{q}, q) = \dot{I}(u; \bar{s} \mid c) = \dot{I}(u; \bar{s} \mid c) - \dot{I}(u; c \mid c) = \dot{I}(u, c; \bar{s} \mid c) \le \dot{I}(s; \bar{s} \mid c). \quad (4.13)$$

Отже, цілком логічно, що величина ППЗ дорівнює взаємній інформації між контейнером-результатом s і спотвореною його версією $\bar{s}$ за умови, що відправникові й одержувачеві прихованої інформації є відомим порожній контейнер c.

Для практичних систем захисту інформації, якщо секретним ключем стеганосистеми є опис використовуваного контейнера, виникають дві проблеми. По-перше, одержувач повинен знати контейнер-оригінал, що обмежує можливу область застосування таких стеганосистем. По-друге, відправник і одержувач прихованих повідомлень повинні використовувати секретну інформацію (ключі) дуже великого обсягу, що не завжди є зручним на практиці.

4.3.2. Властивості прихованої пропускної здатності стеганоканалу

Розглянемо властивості ППЗ, наведені у [5]. Прихована пропускна здатність є функцією аргументів A_1 і A_2, що зручно виразити у вигляді $B(A_1, A_2)$, і вона характеризується наступними властивостями:

- величина $B(A_1, A_2)$ монотонно зростає при збільшенні рівня спотворення, викликаного кодуванням (A_1), і монотонно зменшується при зростанні спотворення, спричиненого атакуючим впливом (A_2);
- функція $B(A_1, A_2)$ є опуклою за аргументом A_2;

- величина $B(A_1, A_2)$ є обмеженою згори ентропією спотвореної стеганограми $\bar{s}$ і ентропією контейнера c:

$$B(A_1, A_2) \le \max_{\bar{q} \in \bar{\Psi}} \min_{q \in \Psi} H(\bar{s}) \le H(s) \le H(c) \le log\,|\mathbf{C}|^{1};$$

- величина $B(0, A_2) = 0$ для будь-яких значень атакуючого спотворення A_2, оскільки $A_1 = 0$ означає, що $s = c$, тобто контейнер-оригінал повністю збігається з контейнером-результатом (жодної прихованої інформації не передається);
- якщо є припустимим достатньо велике атакуюче спотворення A_2, то для будь-якого значення спотворення A_1 може бути побудована атака порушника, в якій стеганопослідовність $\bar{s}^N$ формується незалежно від s^N. Отже, у послідовності $\bar{s}^N$ усунуті всі сліди прихованого повідомлення і ППЗ дорівнює нулю для будь-яких значень спотворення кодування A_1. Таким чином, якщо атакуючий має можливість заглушати КПЗ необмежено потужною завадою, то він гарантовано зруйнує усі стеганограми, що передавалися цим каналом. Але у багатьох практичних випадках стеганографічного обміну в порушника або відсутній такий енергетичний потенціал, або ж ним неможливо скористатися у повній мірі.

4.3.3. Коментарі отриманих результатів

Наведемо представлені у [5] висновки з теореми 4.1 і коментарі властивості прихованої пропускної здатності.

1) Теорема 4.1 стверджує, що існування теоретичної можливості прихованого безпомилкового передавання інформації і теоретичної можливості протидії цьому зводиться до обчислення величини ППЗ B за відомих стратегій сторін і порівняння її з необхідною швидкістю передавання прихованої інформації R. Якщо ППЗ виявиться менше за

---

[1] Дана властивість є очевидною, оскільки ППЗ не може бути більше за ентропію спотвореного контейнера-результату $\bar{s}$. У свою чергу, через можливі втрати інформації внаслідок атакуючого впливу, величина $H(\bar{s})$ не може бути більше за ентропію контейнера s, а $H(s)$ через можливу втрату інформації при вбудовуванні прихованих повідомлень не може перевищувати ентропію $H(c)$ порожнього контейнера c. З теорії інформації відомо, що ентропія джерела не може перевищувати логарифм від потужності його алфавіту [70, 115]. Оскільки найчастіше використовуються контейнери у вигляді істотно надлишкових мультимедійних даних, то для них виконуватиметься нерівність $H(c) << log\,|\mathbf{C}|$, що істотно зменшує можливе значення ППЗ. Отже, для такої стеганосистеми чим ближчими є характеристики дискретних контейнерів до розподілу Бернуллі (або неперервних контейнерів до розподілу Гаусса), тим більша величина ППЗ може бути досягнута.

необхідну швидкість, то навіть теоретично не існує способу передавання прихованих повідомлень без спотворень і задача атакуючого щодо руйнування довільних стеганосистем буде такою, що гарантовано вирішуватиметься.

Оптимальна атака порушника полягає у внесенні такого спотворення A_2, за якого величина ППЗ є меншою за необхідну швидкість передавання прихованих повідомлень. Оптимальна стратегія приховуючого інформацію зводиться до обрання такого кодування і такої величини викликаного ним спотворення A_1, при яких з урахуванням спотворення A_2 необхідна швидкість безпомилкового передавання не перевищуватиме ППЗ. Це означає, що теоретично такий спосіб безпомилкового передавання існує. Однак теоретична можливість ще не означає, що передавальна сторона буде здатною реалізувати її на практиці. Наприклад, розроблювач стеганосистеми може не знати оптимальних принципів її побудови (вони ще не відкриті), або через обмеженість в обчислювальних ресурсах він не в змозі собі дозволити оптимальну обробку, або вимоги до своєчасності доставки прихованих повідомлень обмежують довжину N блока кодування тощо.

Таким чином, успіх будь-якої зі сторін протиборства в остаточному підсумку визначатиметься співвідношенням між швидкістю передавання R і величинами спотворення A_1 і A_2 контейнера, у якому приховується інформація.

Розглянута теорема інформаційного прихування при активній протидії порушника нагадує фундаментальну теорему Шеннона [60, 115], у якій стверджується, що існує спосіб безпомилкового передавання повідомлень по каналу зв'язку з завадами, якщо швидкість передавання є меншою за пропускну здатність каналу, і неможлива достовірна передача із швидкістю, більшою за пропускну здатність цього каналу. Шеннон також показав, що існують залежності між відношенням потужності корисного сигналу до потужності завад у каналі зв'язку та величиною швидкості безпомилкового передавання повідомлень цим каналом. Аналогічно цьому, в інформаційно-приховуючому протиборстві теж існують подібні залежності між відношенням величини спотворення кодування A_1 до величини спотворення атакуючого впливу A_2 та величиною швидкості безпомилкового передавання прихованих повідомлень стеганоканалом. Але, при своїй зовнішній подібності, у задач відкритого та прихованого передавання є істотні відмінності. Відкритий зв'язок здійснюється в умовах впливів випадкових завад у каналі зв'язку, а передавання стеганограм повинна бути забезпечена навіть за умови оптимізованої навмисної протидії активного порушника.

2) Розглянемо зв'язок задачі інформаційного приховання із задачею захисту інформації від перехоплення в каналі зв'язку, який прослуховується. Ефективний метод захисту інформації від читання порушником, що, у підсумку, заклав основу *теорії кодового зашумлення*, був запропонований Авраамом Вайнером *(A. D. Wyner)* [5, 71–73]. Відправник дискретних повідомлень здійснює їх випадкове надлишкове кодування і передає перетворені повідомлення одержувачеві основним каналом зв'язку. Порушник спостерігає їх у підслуховуючому каналі, що є відведенням від основного. Випадкове кодування побудовано таким чином, що якщо у підслуховуючому каналі є помилки, то при декодуванні вони розмножуються й надійно спотворюють захищувану інформацію.

Метод кодового зашумлення призначений для систем зв'язку, основний канал яких є безпомилковим. Наприклад, основний канал утворений на основі волоконно-оптичної лінії, а порушник намагається вести розвідку по каналах побічного електромагнітного випромінення і наведень, в яких у силу їхньої природи існує велика кількість завад. Відзначимо, що порушник знає опис системи кодового зашумлення, що не використовує секретної ключової інформації (тобто спосіб захисту — некриптографічний).

Підслуховуючий канал характеризується секретною ПЗ, яка являє собою максимальну швидкість безпомилкового передавання основним каналом за умови, що невизначеність для перехоплювача є максимальною (невизначеність захищуваних повідомлень дорівнює ентропії цих повідомлень). Однак, якщо підслуховуючий канал є менш зашумленим, аніж основний, то секретна ПЗ дорівнює нулю.

У завданні інформаційного приховання атакуючий здатний на більше, аніж звичайний перехоплювач у підслуховуючому каналі, оскільки після перехоплення захищуваного повідомлення він навмисно спотворює основний канал. Тому основний канал передавання не менш зашумлений, аніж підслуховуючий. Отже, у завданні інформаційного приховання з активним порушником секретна ПЗ дорівнює нулю.

3) Обрання змінної $\boldsymbol{u}$ незалежно від контейнера $\boldsymbol{c}$, як це робиться в системі ЦВЗ [5], є в загальному випадку неоптимальним. Аналіз виразу (4.10) показує, що швидкості безпомилкового передавання в цьому випадку обмежені згори величиною $\boldsymbol{I}(\boldsymbol{u};\ \bar{\boldsymbol{s}} \mid \mathbf{k})$.

4) Нехай виконується умова $\boldsymbol{A}_2 \geq \boldsymbol{A}_1$. Якщо атакуючому відомий опис контейнера $\boldsymbol{c}^N$, то оптимальна атака полягає лише у формуванні спотвореного контейнера у вигляді $\bar{\boldsymbol{s}}^N = \boldsymbol{c}^N$. У цьому випадку вихідний сигнал після атаки не містить жодних слідів повідомлення і ППЗ дорівнює нулю. На практиці це може означати наступне: якщо по-

рушникові відомий оригінал захищуваного від піратського копіювання медіа-продукту, то жодні стеганосистеми не захистять авторські чи майнові права виробників цього продукту.

Розглянемо потенційно сильну атаку, під час якої атакуючий намагається сконструювати досить близьку до оригіналу оцінку контейнера c^N. Якщо він здатний синтезувати спотворену стеганопослідовність s таку, що $H(\tilde{s} \mid c) < \xi$, то ППЗ обмежуватиметься згори величиною

$$\dot{I}(u;\ \tilde{s} \mid k) - \dot{I}(u;\ c \mid k) = \dot{I}(u;\ c, \tilde{s} \mid k) - \dot{I}(u;\ c \mid \tilde{s},\ k) - \dot{I}(u;\ c, s \mid k) - \quad (4.14)$$
$$- \dot{I}(u;\ s \mid c, k) \le \dot{I}(u; s \mid c, k) \le H(\tilde{s} \mid c, k) \le H(\tilde{s} \mid c) < \xi,$$

для будь-якого u. Отже, величина ППЗ стеганоканалу $B(A_1, A_2) < \xi$.

Таким чином, якщо порушник здатен сформувати досить точну оцінку контейнера (іншими словами, виконується нерівність

$$H(\tilde{s} \mid c) < \xi,$$

де величина ξ досить мала), то величина ППЗ обмежена цією малою величиною. На практиці це означає, що маючи заповнений контейнер, порушник може спробувати відтворити з нього контейнер-оригінал (можливо, з деякою припустимою похибкою), з якого вже буде вилучене приховане повідомлення (ця проблема є особливо актуальною у сфері захисту медіа-даних за допомогою ЦВЗ).

4.4. Двійкова стеганосистема

Визначимо величину ППЗ стеганосистеми, у якої алфавіт прихованих повідомлень, контейнерів, ключів і стеганограм є двійковим: $m = c = k = s = \{0; 1\}$ [5, 61, 115].

Нехай контейнер c формується джерелом Бернуллі з параметром $p = 0{,}5$ (тобто двійкові символи послідовності контейнера є рівноймовірними і незалежними один від одного). Функція спотворення $d_1 = d_2$ описується відстанню Геммінга: $d(x, y) = 0$, якщо $x = y$ і $d(x, y) = 1$, якщо $x \neq y$.

Опис контейнера є секретним ключем стеганосистеми ($k = c$) і є відомим одержувачеві. Нехай стеганограми формуються у вигляді $s = c \oplus z$, де операція «$\oplus$» є підсумовуванням за модулем 2. Очевидно, що змінна z теж матиме розподіл Бернуллі і відображатиме собою приховане повідомлення m зі спотворенням A_1.

Спотворення A_1 означає, що кожен символ двійкової послідовності z відрізняється від відповідного символу двійкової послідовності m з імовірністю A_1. Перетворення повідомлення m на послідовність z виконується передавальною стороною з використанням кодера зі

спотворенням A_1. Порушник обробляє стеганограму накладенням на неї двійкової шумової послідовності a, в якій одиничний символ породжується з імовірністю A_2. У свою чергу одержувач підсумовує спотворену стеганограму $\breve{s}$ з двійковою послідовністю c за модулем 2, і з отриманої в такий спосіб двійкової послідовності $\breve{z}$ декодує прийняте приховане повідомлення $\breve{m}$.

Особливістю представленої стеганосистеми є те, що при вбудовуванні нею повідомлення m останнє спотворюється з імовірністю спотворення A_1 і це спотворення дорівнює спотворенню кодування стеганограми. Описана стеганосистема зображена на рис. 4.2.

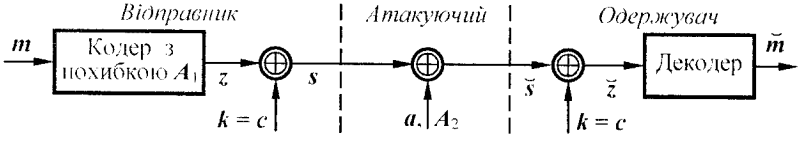


Рис. 4.2. Структурна схема двійкової стеганосистеми

Твердження 4.3

Для двійкової стеганосистеми при рівнях спотворень $A_1, A_2 \le 0.5$, прихована пропускна здатність визначається як

$$B = \underline{H}(A_1 * A_2) - \underline{H}(A_2), \qquad (4.15)$$

де $\underline{H}(t) = -t \cdot log(t) - (1-t) \cdot log(1-t)$; $A_1 * A_2 = A_1 \cdot (1-A_2) + A_2 \cdot (1-A_1)$.

Для даної стеганосистеми змінну u можна формувати як $u = s$ або $u = z$, причому обидва варіанти можуть бути оптимальними, оскільки в якості операції вбудовування використовується операція підсумовування за модулем 2 [5].

Оптимальна атака порушника визначається у вигляді $\breve{s} = s \oplus a$, де a — випадкова двійкова послідовність, розподіл якої описується законом Бернуллі, а ймовірність появи одиничного символу дорівнює A_2.

За рівнів спотворень $A_1 \ge 0{,}5$ і $A_2 < 0{,}5$ ППЗ дорівнює $B = 1 - \underline{H}(A_2)$. Якщо $A_2 \ge 0.5$, то ППЗ дорівнюватиме нулю.

Необхідно зауважити, що при $A_1 = 0{,}5$ і $A_2 < 0{,}5$ ППЗ не дорівнює нулю незалежно від значення A_2. Це пояснюється тим, що при перетворенні прихованого повідомлення m у послідовність z спотворення не є рівноймовірним: особа, що приховує інформацію, може обрати такий розподіл помилок A_1, при якому мінімізуватиметься зміна

повідомлення m. Для $A_2 = 0{,}5$ ППЗ дорівнюватиме нулю при будь-яких значеннях A_1. Неважко помітити, що у цьому випадку вихід $\breve{s}$ каналу зв'язку не залежить від його входу s, що означає розрив каналу зв'язку. А якщо при обриві каналу зв'язку не можлива передача інформації по відкритому каналу зв'язку, то тим більше не можлива й передача прихованим каналом, який утворено на основі відкритого.

Застосуємо наслідок теореми 4.1 для аналізу двійкової стеганосистеми. Нехай $z = c \oplus s$, $a = s \oplus \breve{s}$. Цільова платіжна функція має вигляд $\dot{I}(s; \breve{s} \mid c)$. Приймемо, що $A_1, A_2 \le 0{,}5$.

Крок 1. Зафіксуємо $q(\breve{s} \mid s)$. Для всіх $q \in \psi$, одержимо:

$$\dot{I}(s; \breve{s} \mid c) \overset{(a)}{=} H(\breve{s} \mid c) - H(\breve{s} \mid s, c) \overset{(b)}{=}$$
$$\overset{(b)}{=} H(\breve{s} \mid c) - H(\breve{s} \mid s) = H(\breve{s} \oplus c \mid c) - H(a) = H(z \oplus a \mid c) - \underline{H}(A_2) \overset{(c)}{\le}$$
$$\overset{(c)}{\le} H(z \oplus a) - \underline{H}(A_2) \overset{(d)}{\le}$$
$$\overset{(d)}{\le} \underline{H}(A_1 * A_2) - \underline{H}(A_2),$$

де рівність (a) справедлива відповідно до визначення умовної взаємної інформації; рівність (b) виконується завдяки тому, що $c \to s \to \breve{s}$ є марківським ланцюгом. Нерівність у (c) справедлива, оскільки умова зменшує ентропію, а рівність у (c) досягається тоді і тільки тоді, коли $z \oplus a$ і, таким чином, z є незалежною від c. Нерівність у (d) буде справедливою, оскільки z і a є незалежними (в силу того, що $z \to s \to a$ формує марківський ланцюг і $P[z = 1] \le A_1$), а рівність у (d) досягається, якщо змінна z має розподіл Бернуллі з дисперсією A_1. Розподіл $p(s \mid c)$ задовольняє обом несуворим нерівностям і тому максимізує значення $\dot{I}(s; \breve{s} \mid c)$.

Крок 2. Зафіксуємо $p(s \mid c)$. Мінімізуватимемо $\dot{I}(s; \breve{s} \mid c)$ над $q(\breve{s} \mid s)$. При визначеному раніше розподілі $p(s \mid c)$, послідовності z і s є незалежними. Оскільки $z \to s \to a$ формує марківський ланцюг, то z і a також є незалежними. Тоді маємо

$$\dot{I}(s; \breve{s} \mid c) = \dot{I}(s \oplus c;\ \breve{s} \oplus c \mid c) = \dot{I}(z; z \oplus a \mid c) = H(z) - H(z \mid z \oplus a, c) \overset{(e)}{\ge}$$
$$\overset{(e)}{\ge} H(z) - H(z \mid z \oplus a) = \dot{I}(z; z \oplus a) \overset{(f)}{\ge} \underline{H}(A_1 * A_2) - \underline{H}(A_2),$$

де справедливість нерівності (e) зумовлена тим, що умова зменшує ентропію, а нерівності (f) — оскільки z і a є незалежними і $P[a = 1] \le A_2$ *(що стає рівністю, коли a — це змінна з бернуллівським розподілом та ймовірністю одиничного символу A_2).*

Розглянута двійкова стеганосистема схожа на систему шифрування з однократною підстановкою (потоковий шифр гамування з нескінченною рівноймовірною і незалежною шифруючою гамою). При незалежному і рівноймовірному характері послідовності c виконується рівність $H(z) = H(z \mid s)$ [115], що означає, що дана система задовольняє вимозі щодо ідеальних криптосистем [60], коли перехоплення й аналіз криптограми s не дає атакуючому жодної інформації стосовно захищуваного повідомлення z. Однак ця двійкова система задовольняє також вимозі щодо ідеальних стеганосистем: розподіли $p(c)$ і $p(s)$ ідентичні, тому для порушника неможливо визначити, чи належать перехоплені ним дані до розподілу $p(c^N)$ порожніх контейнерів або ж до розподілу $p(s^N)$ контейнерів-результатів [15]. Але при цьому зауважується, що в розглянутій стеганосистемі передбачається опис контейнерів і, відповідно, стеганограм розподілом Бернуллі, а це зазвичай не є характерним для реальних систем приховання інформації [5].

Розглянемо приклад двійкової стеганосистеми з вибором $u = z$ [5]. Нехай існує необхідність у прихованому передаванні повідомлення m, що являє собою оцифрований мовний сигнал з кількістю рівнів квантування 8. У загальному вигляді приховане повідомлення може бути представлене у вигляді $\mathbf{M} = \{m_1, m_2, m_3, m_4, \ldots\}$. Нехай перші декілька відліків повідомлення в моменти часу дискретизації t_1, t_2, t_3, t_4, ... приймають десяткові значення $m_1 = 0$, $m_2 = 8$, $m_3 = 19$, $m_4 = 80$ (рис. 4.3, *а*). У двійковій формі приховане повідомлення запишемо як

$$m_1 = 0000\,0000_2,\ m_2 = 0000\,1000_2,\ m_3 = 0001\,0011_2,\ m_4 = 0101\,0000_2, \ldots$$

Перетворимо двійкову послідовність $\mathbf{M}$ у двійкову послідовність $\mathbf{Z}$ з похибкою A_1. У двійковій стеганосистемі похибка кодування A_1 обчислюється за метрикою Геммінга. Нехай $A_1 = 1/8$. Отже, для формування послідовності $\mathbf{Z} = \{z_1, z_2, z_3, z_4, \ldots\}$ особа, що приховує інформацію, спотворює восьму частину бітів послідовності $\mathbf{M}$. Для зменшення спотворення прихованого повідомлення йому доцільно спотворювати тільки молодші біти двійкової послідовності $\mathbf{M}$.

Нехай на передавальному боці було обрано послідовність $\mathbf{Z}$ наступного вигляду:

$$z_1 = 0000\,0001_2,\ z_2 = 0000\,1001_2,\ z_3 = 0001\,0001_2,\ z_4 = 0101\,0010_2, \ldots .$$

У десятковому вигляді послідовність $\mathbf{Z}$ зображена на рис. 4.3, *б*.

Припустимо, що за допомогою генератора ПВП було сформовано секретний ключ $\mathbf{K} = \{k_1, k_2, k_3, k_4, \ldots\}$:

$$k_1 = 0100\,1101_2,\ k_2 = 0111\,0010_2,\ k_3 = 0101\,0101_2,\ k_4 = 0101\,1001_2, \ldots$$

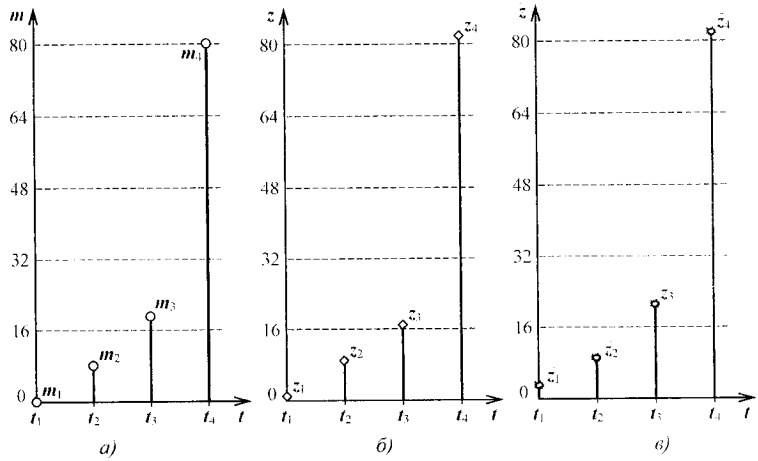


Рис. 4.3. Приклад двійкової стеганосистеми
зі спотвореннями $A_1 = 1/8$ і $A_2 = 1/16$

Відправник за правилом побітового додавання $\mathbf{S} = \mathbf{K} \oplus \mathbf{Z}$ формує контейнер-результат $\mathbf{S} = \{s_1, s_2, s_3, s_4, ...\}$:

$s_1 = 0100\,1100_2$, $s_2 = 0111\,1011_2$, $s_3 = 0100\,0100_2$, $s_4 = 0000\,1011_2$, ...

Нехай спотворення $A_2 = 1/16$. Порушник формує псевдовипадкову двійкову послідовність $\mathbf{A} = \{a_1, a_2, a_3, a_4, ...\}$, в якій імовірність появи одиничних символів складає A_2. Наприклад,

$a_1 = 0000\,0010_2$, $a_2 = 0000\,0000_2$, $a_3 = 0000\,0100_2$, $a_4 = 0000\,0000_2$, ...

Атакуючий вплив являє собою додавання за модулем 2 контейнера-результату $\mathbf{S}$ до шумової послідовності $\mathbf{A}$. Отже, спотворений контейнер $\mathbf{S} = \{\tilde{s}_1, \tilde{s}_2, \tilde{s}_3, \tilde{s}_4, ...\}$ матиме наступний вигляд:

$\tilde{s}_1 = 0100\,1110_2$, $\tilde{s}_2 = 0111\,1011_2$, $\tilde{s}_3 = 0100\,0000_2$, $\tilde{s}_4 = 0000\,1011_2$, ...

Одержувач для формування прийнятого повідомлення $\tilde{\mathbf{Z}}$ додає за модулем 2 послідовність $\tilde{\mathbf{S}}$ до послідовності ключа $\mathbf{K}$:

$\tilde{z}_1 = 0000\,0011_2$, $z_2 = 0000\,1001_2$, $\tilde{z}_3 = 0001\,0101_2$, $\tilde{z}_4 = 0101\,0010_2$, ...

У декодері одержувач з даної послідовності відновлює повідомлення $\mathbf{M}$. У найпростішому випадку $\tilde{\mathbf{M}} = \tilde{\mathbf{Z}}$. Вигляд послідовності $\tilde{\mathbf{M}}$ зображено на рис. 4.3, *в*.

Якщо приховане повідомлення **M** являє собою мовний сигнал, то при зазначених величинах спотворень A_1 і A_2 ступінь наближеності **M** до **M** (зокрема, якість забезпечуваного прихованого телефонного зв'язку для низки телекомунікаційних задач) може бути оцінений як задовільний.

4.5. Підсумки розділу

У даному розділі введене до розгляду одне з ключових понять теорії передачі інформації і, зокрема, стеганографічних систем (як каналів прихованого обміну даними) — пропускна здатність каналу передавання. Під час опрацювання закордонних і вітчизняних літературних джерел виокремлено два основних підходи до оцінки ПЗ стеганосистем:

- підхід, орієнтований на стеганосистеми, в яких приховані повідомлення мають бути безпомилково передані адресатові в умовах активної протидії порушника (при цьому враховується, що крім спотворювань контейнера, що виникають за результатом вбудовування до нього конфіденційних даних, є ймовірними і його навмисні спотворення з боку активного порушника і/або спотворення, викликані випадковими завадами, що діють у каналі зв'язку);
- підхід, орієнтований на стеганосистеми, в яких реалізується приховане передавання апріорно невідомої одержувачеві інформації, причому пасивний порушник намагається в процесі спостереження виявити факт наявності каналу прихованого зв'язку і, в разі успіху, прагне розкрити зміст прихованих даних.

В рамках першого підходу наведено основні завдання інформаційного приховання у випадку активної протидії зловмисника; описано приховуюче перетворення, виконуване при вбудовуванні інформації до контейнера, і атакуючий вплив, здійснюваний порушником для протидії прихованому передаванню; розглянута основна теорема інформаційного приховання при активній протидії порушника; наведений приклад визначення величини прихованої пропускної здатності двійкової стеганографічної системи.

На основі цього розглянуто основні властивості прихованої пропускної здатності стеганоканалу і прокоментовано отримані результати, що дозволило закласти обґрунтовану теоретичну базу для розробки систем стеганографічного приховання конфіденційної інформації.

Розділ 5. СТЕГАНОГРАФІЧНІ МЕТОДИ ПРИХОВАННЯ ДАНИХ

5.1. Вступні положення

У даному розділі розглядаються стеганографічні методи приховання даних для різних мультимедійних типів інформаційного середовища в якості стеганоконтейнерів. При цьому основну увагу приділено проблемі практичної реалізації розгляданих методів — з використанням сучасних засобів обчислювальної техніки і популярного програмного забезпечення.

Згідно з [3, 112], під час розгляду методів позначатимемо літерою **C** мультимедійний контейнер, що являє собою послідовність елементів c_i довжиною l_C. У випадку використання в якості контейнера файлу цифрового звуку це буде кількість наявних часових відліків, для файлу цифрового зображення — послідовність, отримана шляхом векторизації (розгортання у вектор) масиву усіх пікселів зображення.

Для двійкових масивів контейнерів c_i можуть набувати значень 0 або 1; для квантованого зображення або звуку, для яких кількість бітів, якими кодується яскравість окремого пікселя або інтенсивність звучання окремого часового відліку, дорівнює 8, — змінюватись у діапазоні від 0 до 255 ($2^8 = 256$ градацій); для звуку, відліки якого кодуються 16-ма бітами, — від –32768 до 32767 ($2^{16} = 65536$ градацій); для текстів c_i — це символ кодової таблиці, код якого, наприклад, може приймати значення від 0 до 255.

Аналогічно, позначатимемо літерою **S** заповнений контейнер — послідовність елементів s_j довжиною l_S, а літерою **M** — повідомлення довжиною l_M, що підлягає стеганографічному прихованню. Також, якщо про інше не буде застережено, вважатимемо, що $m_n \in \{0; 1\}$.

5.2. Класифікація стеганографічних методів приховання даних

Переважна більшість методів комп'ютерної стеганографії (КС) базується на двох ключових принципах [3]:

- файли, що не вимагають абсолютної точності (наприклад, файли з зображенням, аудіоінформацією тощо), можуть бути видозмінені (звичайно ж, лише до певного ступеня) без втрати своєї функціональності;

- органи відчуттів середньостатистичної людини не здатні надійно розрізняти незначні зміни у модифікованих таким чином файлах та/або відсутній спеціальний інструментарій, який би був спроможним виконати таку задачу.

Оскільки в комп'ютерній стеганографії розглядаються медіасередовища, підтримувані переважною більшістю сучасних засобів обчислювальної техніки та інфокомунікаційних мереж, то все інформаційне середовище, зрештою, може бути представлене у цифровому вигляді [3]. В основі базових підходів до реалізації відомих методів КС в рамках того чи іншого мультимедійного середовища лежить виокремлення малозначущих фрагментів останнього і підміна існуючої в них інформації на ту інформацію, яку необхідно приховати. Іншими словами, неважливі для певних кадрів даних фрагменти замінюються (заміщуються) на складові приховуваної інформації відповідно до використовуваного алгоритму чи методики. Під кадром інформаційного середовища в даному випадку розуміється певна його частина, виділена за характерними ознаками (семантичними характеристиками). Наприклад, кадром може бути обране якесь окреме зображення, звуковий файл, *web*-сторінка і т. п.

Для існуючих методів комп'ютерної стеганографії може бути введена наступна класифікація (рис. 5.1) [3, 5, 112].

Як уже зазначалося у розділі 2, ***за способом обрання контейнера*** розрізнюють сурогатні (так звані неповноцінні або ерзац-методи), селективні та конструюючі методи стеганографії [3].

В *сурогатних* (*безальтернативних*) методах можливість вибору контейнера повністю відсутня, отож, для приховання повідомлення обирається перший-ліпший контейнер, що трапився, — ерзац-контейнер, — що у більшості випадків навряд чи виявиться оптимальним для приховання повідомлення заданого формату.

У *селективних* методах КС передбачається, що приховане повідомлення повинне відтворювати певні спеціальні статистичні характеристики шуму контейнера. Для цього генерують велику кількість альтернативних контейнерів, з наступним обранням (шляхом відбраковування) найоптимальнішого з них для конкретного повідомлення. Окремим випадком такого підходу є обчислення деякої хеш-функції для кожного контейнера. При цьому для приховання повідомлення обирається той контейнер, хеш-функція якого найкращим чином збігається зі значенням хеш-функції повідомлення (тобто у цьому випадку стеганограмою є обраний контейнер).

У *конструюючих* методах стеганографії контейнер генерується самою стеганосистемою. При цьому існують декілька варіантів реалізації. Так, наприклад, шум контейнера може імітуватися прихованим

повідомленням. Це здійснюється за допомогою процедур, що не лише кодують приховане повідомлення «під шум», але й зберігають модель початкового шуму. У граничному випадку за моделлю шуму може будуватися ціле повідомлення. Прикладом, зокрема, може слугувати метод, реалізований у програмі *MandelSteg* [74], що в якості контейнера генерує фрактал Мандельброта (*Mandelbrot fractal*), або ж апарат функцій імітації [67].

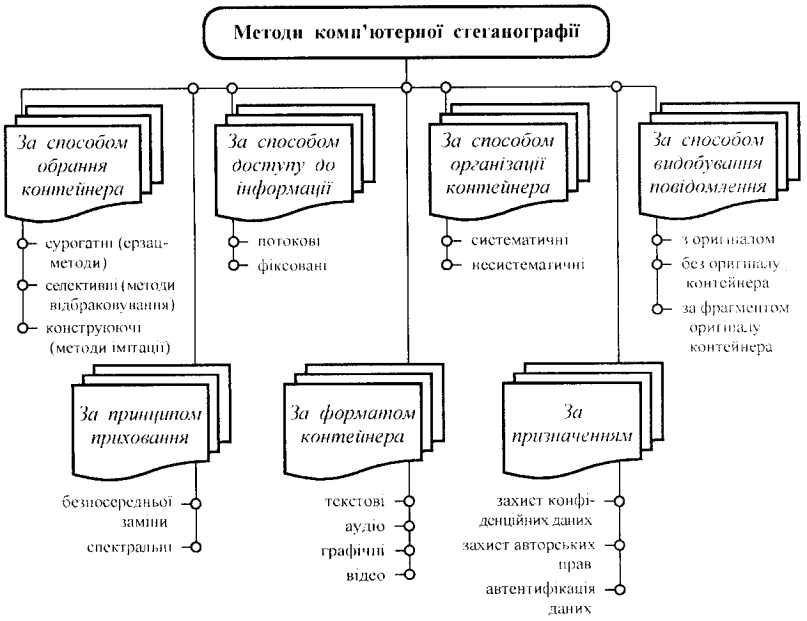


Рис. 5.1. Класифікація методів комп'ютерної стеганографії

За способом доступу до прихованої інформації розрізнюють методи для *потокових* (*безперервних*) контейнерів і методи для фіксованих (*обмеженої довжини*) контейнерів (докладніше див. підрозділ 2.3).

За способом організації контейнери, подібно корегувальним кодам, можуть бути *систематичними* і *несистематичними* [3, 115]. У бітовій структурі перших можна однозначно виділити конкретні місця, де приховано інформаційні біти власне повідомлення, і фрагменти, інформація яких слугує свого роду шумом, що маскує собою стеганограму (як, наприклад, у класичному методі найменшого значущого

біта). У випадку ж несистематичної організації контейнера такий поділ неможливий — у цьому разі для видобування бітів прихованої інформації необхідно обробляти вміст усього контейнера.

За використовуваним принципом приховання методи КС поділяють на два основних класи: *безпосередньої заміни* і *спектральні*. Якщо перші, використовуючи надлишковість інформаційного середовища контейнера у просторовій (для зображення чи тексту) або часовій (для звуку) області, полягають в заміні малозначимої частини контейнера бітами секретного повідомлення, то другі для приховання даних використовують вже спектральне представлення елементів контейнера (наприклад, коли до різних коефіцієнтів масивів-результатів дискретних косинусних перетворень, перетворень Фур'є, Карунена-Лоева, Адамара, Гаара тощо) [5].

Основним напрямком комп'ютерної стеганографії є використання властивостей саме надлишковості пустого контейнера. Але при цьому треба зважати й на те, що в результаті приховання інформації відбувається викривлення деяких статистичних властивостей контейнера і/або порушення його структури. Для зменшення демаскувальних ознак на цей факт слід обов'язково зважати.

До окремої групу можна також виділити методи, що ***використовують спеціальні властивості форматів представлення файлів*** [3]:

- зарезервовані для розширення поля файлів, які зазвичай заповнюються нулями і зазвичай не враховуються програмою;
- спеціальне форматування даних (зсування слів, речень, абзаців або обирання визначених позицій символів);
- використання незадіяних ділянок пам'яті на магнітних, оптичних чи напівпровідникових носіях;
- видалення файлових заголовків-ідентифікаторів тощо.

В основному, для таких методів характерні низький ступінь прихованості, замала пропускна здатність і слабка продуктивність.

За призначенням розрізнюють стеганографічні методи власне для *прихованого передавання* (*зберігання*) даних і методи для приховання даних у цифрових медіа-об'єктах з метою можливості доведення своїх прав на них згодом.

За типами використовуваного контейнера виділяють стеганографічні методи з текстовими, графічними, звуковими та аудіовізуальними контейнерами.

Надалі пропонується розглянути докладніше стеганографічні методи приховання даних у статичних зображеннях, в аудіосигналах і текстових файлах.

5.3. Приховування даних у статичних зображеннях

Більшість досліджень у сфері комп'ютерної стеганографії присвячена використанню в якості контейнерів саме зображень. Це зумовлено наступними причинами:

- існуванням практичної необхідності захисту цифрових фотографій, різноманітних графічних документів чи відео від протизаконного тиражування й розповсюдження;
- відносно великим обсягом цифрового представлення зображень, що дозволяє вбудовувати ЦВЗ великого обсягу або ж підвищувати стійкість такого вбудовування;
- заздалегідь відомим (фіксованим) розміром контейнера, відсутністю обмежень, що накладаються вимогами прихування у реальному часі;
- наявністю в більшості реальних зображень текстурних областей, що мають шумову структуру і найкращим чином підходять для вбудовування секретної інформації;
- слабкою чутливістю людського ока до незначних змін кольорів зображення, його яскравості, контрастності, вмісту в ньому шуму, спотворень поблизу контурів;
- зрештою, вельми добре розробленими останнім часом методами цифрової обробки зображень.

Але, як зазначається в [5], остання причина породжує й значні труднощі у забезпеченні стійкості ЦВЗ: чим більш досконалими стають методи стиснення, тим менше лишається можливостей для вбудовування сторонньої інформації.

Розвиток теорії та практики алгоритмів стиснення зображень призвів і до зміни уявлень про техніку вбудовування ЦВЗ. Якщо спочатку пропонувалося вбудовувати інформацію в незначущі біти для зменшення візуальної помітності, то сучасний підхід, навпаки, полягає у вбудовуванні ЦВЗ до найбільш суттєвих ділянок і областей зображень, руйнування яких призводитиме до повної деградації захищених в такий спосіб контейнерів. Тому цілком зрозумілою є необхідність врахування стеганоалгоритмами не лише алгоритмів стиснення зображень, але й властивостей зорової системи людини (ЗСЛ).

5.3.1. Властивості ЗСЛ у контексті побудови стеганоалгоритмів

Усі властивості ЗСЛ можна умовно поділити на дві групи: *низькорівневі* (тобто «фізіологічні») і *високорівневі* («психофізіологічні») [75,76].

Майже до середини 90-х років дослідники брали до уваги, головним чином, низькорівневі властивості зору. В останні ж десятиліття позначилася стійка тенденція побудови стеганоалгоритмів з урахуванням ще й високорівневих характеристик ЗСЛ.

Виокремлюють три найважливіші ***низькорівневих властивості***, що впливають на помітність стороннього шуму в зображенні:

- чутливість до зміни яскравості (контрасту) зображення;
- частотна чутливість;
- ефект маскування.

На рис. 5.2 представлено залежність мінімального *контрасту* $\Delta i / i$ від *яскравості* i.

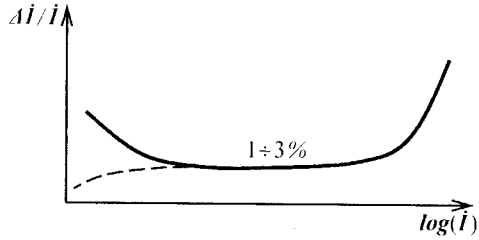


Рис. 5.2. Чутливість до зміни контрасту і поріг нерозрізненості ΔI

Як видно, для середнього діапазону зміни яскравості, контраст є приблизно постійним, тоді як для малих і великих яскравостей значення порогу нерозрізненості (Δi) зростає. Встановлено, що для середніх значень яскравості $\Delta i \approx (0.01 \div 0.03) \cdot i$.

Крім того, у [5] відзначено, що результати новітніх досліджень суперечать «класичній» теорії і показують, що при малих значеннях яскравості поріг нерозрізненості зменшується, тобто ЗСЛ стає більш чутливою до шуму в цьому діапазоні.

Частотна чутливість ЗСЛ проявляється в тому, що людина є набагато більше сприйнятливою до низькочастотного (НЧ) шуму, аніж до високочастотного (ВЧ). Головним чином це пов'язане з нерівномірністю амплітудно-частотної характеристики ЗСЛ.

Елементи ЗСЛ розщеплюють відеосигнал, що надходить, на окремі складові, кожна з яких збуджує нервові закінчення ока через низку субканалів. Виокремлювані оком складові мають різні просторові й частотні характеристики, а також різну просторову орієнтацію (горизонтальну, вертикальну, діагональну) [77]. У випадку одночасного впливу на око двох складових з подібними між собою характеристиками збуджуються одні й ті самі субканали. Це призводить до так званого

ефекту маскування, що полягає у збільшенні порогу виявлення зорового сигналу в присутності іншого сигналу, що має аналогічні характеристики. Саме тому адитивний шум є набагато помітнішим на НЧ (однотонних) ділянках зображення, аніж на високочастотних, — в останньому випадку спостерігається маскування. Найбільш потужно даний ефект проявляється, коли обидва сигнали мають однакову орієнтацію і місце розташування [5, 75].

Частотна чутливість тісно пов'язана з яскравісною. Відомий також і вираз [119] для визначення порогу маскування на основі відомої яскравісної чутливості, що, в принципі, дозволяє знайти метрику спотворення зображення, яка враховувала б властивості ЗСЛ. Такого типу математичні моделі, зокрема, досить добре розроблені для випадку квантування коефіцієнтів дискретного косинусного перетворення зображення, що ефективно застосовується в стандарті *JPEG*.

Ефект маскування у *просторовій області* може бути пояснений шляхом побудови ймовірнісних моделей зображення. При цьому останнє представляється у вигляді марківського випадкового поля, розподіл імовірностей якого описується, наприклад, узагальненим гаусівським законом.

Запропонована у [5] узагальнена схема вбудовування даних до зображення базується на використовуваній у *JPEG* і *MPEG* кодуваннях психовізуальній моделі ЗСЛ і полягає в наступному:

1) за допомогою орієнтованих смугових фільтрів виконується фільтрація зображення, за результатами якої одержується розподіл енергії за частотно-просторовими компонентами;
2) обчислюється поріг маскування на основі знання локальної величини енергії;
3) у кожній частотно-просторовій компоненті контейнера значення енергії приховуваного елемента повідомлення масштабується таким чином, щоб воно не перевищувало поріг маскування.

Зазначимо, що подібна схема вбудовування в тому чи іншому вигляді використовується чи не в усіх стеганографічних алгоритмах, на основі як графічних, так і звукових контейнерів.

Високорівневі властивості ЗСЛ при побудові стеганографічних алгоритмів враховуються порівняно рідко [5]. Особливістю таких властивостей у порівнянні з низькорівневими є те, що, на відміну від останніх, вони є свого роду побічним продуктом, бо проявляються «вторинно»: обробивши первинну інформацію від ЗСЛ, людський мозок надсилає зоровій системі ті чи інші команди щодо її додаткового «налаштування» під зображення. Основним з високорівневих властивостей є наступні:

- *чутливість до контрасту* — більш контрастні ділянки зображення й істотні перепади яскравості привертають до себе й більшу увагу;
- *чутливість до розміру* — порівняно з меншими, більші за розміром ділянки зображення є «помітнішими» (разом з тим, існує певний поріг насичення, коли подальше збільшення розміру вже не відіграє ролі);
- *чутливість до форми* — довгі й тонкі об'єкти привертають значно більшу увагу, аніж закруглені й однорідні;
- *чутливість до кольорів* — деякі кольори (наприклад, червоний) є «помітнішим» за інші, і цей ефект посилюється, якщо фон заднього плану відрізняється від кольорів фігур на ньому;
- *чутливість до місця розташування* — у першу чергу людина схильна розглядати центр зображення, а вже потім — периферію (також уважніше розглядаються фігури переднього плану, аніж заднього);
- *чутливість до зовнішніх подразників* — рух очей споглядача залежить від конкретної обстановки, від отриманих ним перед переглядом або під час нього інструкцій, додаткової інформації тощо.

Останнім часом створено значну кількість методів стеганографічного приховування даних у цифрових зображеннях, що дозволяє провести їхню класифікацію і виділити наступні узагальнюючі групи [3]:

- методи заміни в просторовій (часовій) області;
- методи приховування в частотній області зображення;
- широкосмугові методи;
- статистичні (стохастичні) методи;
- методи спотворення;
- структурні методи.

Далі у розділі розглядаються особливості, характерні для кожної з виділених груп. Паралельно наводяться приклади програмних модулів у системі *Mathcad*, що дозволяють виконати реалізацію того чи іншого методу, а також проміжні і кінцеві результати відповідних стеганографічних перетворень. За наявної на це потреби, до кожного модуля приведено стислі пояснення щодо його функціонування і використаних функцій *Mathcad*. При цьому, звісно ж, найбільш вичерпні пояснення даються до перших розгляданих методів. У подальшому значення функцій, зміст яких було розкрито раніше, не пояснюється.

Повну уяву про можливості і правила використання (синтаксис) типових об'єктів програми *Mathcad* можна отримати, наприклад, з

[25, 26]. Крім того у даній роботі в *додатках А, Б, В, Г* подано стислу інформацію стосовно вбудованих операторів, функцій і директив, а також системних змінних та програмних операторів *Mathcad*.

5.3.2. Приховування даних у просторовій області зображення

Алгоритми, що описуються у підрозділі 5.3, вбудовують дані прихованих повідомлень безпосередньо в область (масив) первинного зображення. Їх перевагою є те, що для здійснення такого вбудовування нема потреби виконувати обчислювально складні і тривалі лінійні перетворення.

Масив пікселів кольорового зображення **C**, що для цілей стеганографічного приховання виступає множиною контейнерів, представлятимемо через дискретну функцію, яка для кожного пікселя зображення (x, y) визначає вектор кольору $c(x, y)$, де саме значення кольору задає трикомпонентний вектор у колірному просторі. Найбільш поширеним способом передавання кольору є модель *RGB (Red-Green-Blue)*, у якій основні кольори — це червоний, зелений і синій, а будь-який інший колір може бути представлений у вигляді зваженої суми основних. Вектор кольору $c(x, y)$ у *RGB*-просторі представляє собою інтенсивність основних кольорів. Повідомлення вбудовується за рахунок маніпуляцій з колірними складовими $\{\boldsymbol{R}(x, y), \boldsymbol{G}(x, y), \boldsymbol{B}(x, y)\}$ або безпосередньо з яскравістю $\boldsymbol{\lambda}(x, y) \in \{1, 2, \ldots, \boldsymbol{Lc}\}$.

Загальний принцип даних методів полягає в заміні надлишкової, малозначущої частини зображення бітами секретного повідомлення, а для видобування повідомлення необхідно знати алгоритм і/або ключ, за якими розміщувалася по контейнеру інформація приховуваного повідомлення.

5.3.2.1. Метод заміни найменш значущого біта

Метод заміни *найменш значущого біта* (НЗБ, *LSB — Least Significant Bit*) є найпоширенішим серед існуючих методів стеганографічного приховання інформації у просторовій області [3, 5, 9, 14, 19, 20, 120].

Наймолодший значущий біт у складі кодової комбінації градації яскравості пікселя зображення, будучи рівним кроку квантування останнього, одночасно є порівнянним і з шумом (помилкою) квантування [115]. Відомо, що за достатньої глибини дискретизації зображення по рівнях (а в моделі *RGB*, наприклад, на кожен з каналів припадає по 256 рівнів квантування) людина у більшості випадків не спроможна помітити зміну в цьому біті. Фактично, за своєю сутністю

множина НЗБ зображення є шумом і тому її можна використовувати для перенесення бітів стеганограми. При цьому для зображення в градаціях сірого (коли кожен піксель зображення кодується одним байтом) обсяг вбудовуваних даних може сягати 1/8 від загального об'єму контейнера. Приміром, у зображення розміром 512×512 пікселів (розмір файлу ~256 КіБ) заміною НЗБ можна вбудувати ~32 КіБ інформації. Якщо ж у кожній кодовій комбінації яскравості змінювати не один, а одразу два молодших біта (що також буде практично непомітним на око), то приховану пропускну здатність можна збільшити вдвічі.

Популярність методу заміни НЗБ обумовлена його простотою і тим, що він дозволяє приховувати у відносно невеликих файлах досить великі об'єми інформації (пропускна здатність створюваного прихованого каналу зв'язку може складати від 12,5 до 25%, і при цьому пустий і заповнений контейнери майже не відрізнятимуться один від одного). Зазвичай цей метод працює з растровими зображеннями, що представлені у форматі без стиснення із втратами (наприклад, *GIF*, *BMP*, *PNG*, *PCX* тощо) [3][II]. Основним недоліком методу заміни НЗБ є його вкрай незадовільна стеганографічна стійкість до атак пасивного та активного порушників і, як наслідок, висока чутливість до найменших спотворень контейнера. У певній мірі для послаблення цієї чутливості додатково може бути застосоване попереднє завадостійке кодування приховуваного повідомлення.

Перед імпортом зображення-контейнера в документ *Mathcad*, його необхідно підготувати у відповідному редакторі і записати у вигляді файлу в поточний (для формованого документа *Mathcad*) каталог роботи (слід зауважити, що для уникнення можливих проблем з підтримкою кирилиці бажано, щоб адреса розміщення файлу на диску, як, власне, й ім'я файлу, не містили кириличних символів). *Mathcad* підтримує такі формати як *BMP*, *JPEG*, *GIF*, *PCX* і *TGA*. Оскільки деякі з них дозволяють зберігати зображення без втрати у якості, то саме їх і доцільно використовувати в ролі носіїв стеганограм.

Розглянемо, наприклад, структуру *BMP*-файлу [121, 122]: останній містить точкове (растрове) зображення і складається з трьох основних розділів: заголовку файлу, заголовку растру і растрових даних. Заголовок файлу містить інформацію про файл (його тип, об'єм і т. п.). До заголовку растру винесено інформацію про ширину і висоту зображення, кількість бітів на піксель, розмір растру, глибину кольору, коефіцієнт

[II] Узагалі, дані формати використовують алгоритми стиснення на зразок кодування довжин серій (*RLE — Run-Length Encoding*) чи Лемпеля-Зіва-Велча (*LZW — Lempel-Ziv-Welch*), але останні є найпростішими і дозволяють відновлювати зображення з точністю до окремого біта.

стиснення тощо. Нас же передусім цікавитимуть растрові дані — інформація про колір кожного пікселя зображення. Колір пікселя визначається сполученням трьох основних колірних складових — червоного, зеленого і синього, кожному з яких відповідає своє значення інтенсивності, яке може змінюватися від 0 до 255. Отже, за кожен з колірних каналів відповідає 8 бітів (1 байт), а глибина кольору зображення в цілому — 24 біти (3 байти).

Крок 1

Імпорт графічного файлу виконується операцією *Picture* з позиції *Insert* головного меню програми. У модулі, що при цьому з'явився, необхідно заповнити шаблон даних у лівому нижньому кутку, для чого в подвійних лапках ввести ім'я файлу (або ж, за необхідності, — повний шлях його розміщення на диску) і натиснути клавішу <*Enter*>.

Приклад повнокольорового зображення, відтвореного за допомогою операції *Picture* наведений на рис. 5.3. Дане зображення має розмір 128×128 пікселів, глибина кольору — 24 біти, розмір файлу: 48 КіБ.

Рис. 5.3. Зображення-контейнер

Для можливості подальшої обробки зображення, необхідно перевести колірні характеристики кожного його пікселя у числову матрицю. Для виконання цієї операції можна застосувати функцію

READRGB("ім'я_файлу"),

що повертає масив, з трьох підмасивів, які, у свою чергу, містять інформацію про розклад повнокольорового зображення на колірні компоненти *R*, *G* і *B*:

C := **READRGB**("C.bmp").

При цьому три зазначені колірні компоненти розміщуються одна за одною у спільному масиві **C** (рис. 5.4, *а*). На рис. 5.4, *б* приведена гра-

фічна інтерпретація масиву **C** у вигляді трьох півтонових зображень для кожної з колірних компонент. При цьому образ ліворуч характеризує попіксельну інтенсивність у зображенні "C.bmp" червоного, середній — інтенсивність зеленого, а той, що праворуч, — інтенсивність синього.

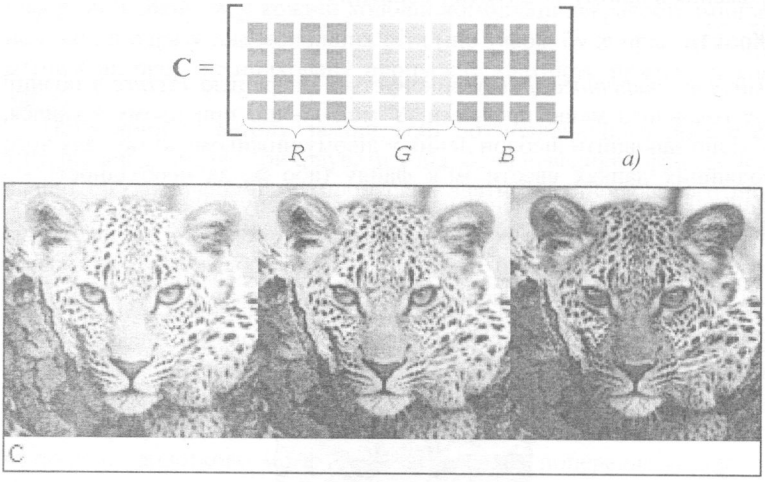


Рис. 5.4. Графічна інтерпретація масиву колірних компонентів контейнера-оригіналу

Для відокремлення колірних складових також можна використати вбудовані функції *Mathcad*, кожна з яких повертає масив, що відповідає певній колірній компоненті графічного файлу:

```
R := READ_RED("C.bmp");
G := READ_GREEN("C.bmp");
B := READ_BLUE("C.bmp").
```

Крок 2

В якості повідомлення, що підлягатиме приховуванню, використаємо, наприклад, перші вісім абзаців зі вступу до даної книги. Текст повідомлення збережемо у файлі "M.txt" поточного для формованого документа *Mathcad* каталогу роботи у такому форматі: тип файлу — *звичайний текст (*.txt)*; кодування — *кирилиця (Windows)*[III]. Імпорт

[III] У принципі ж, існує можливість проведення стеганографічного приховання файлів довільного формату. Єдина умова, що при цьому обов'язково має виконуватися —

текстового файлу можна виконати за допомогою вбудованої функції *Mathcad* **READBİN**("ім'я_файлу", "тип_формату_даних"). У нашому випадку дані будуть представлені множиною 8-бітових беззнакових цілих чисел (байтів):

M := **READBİN**("M.txt", "byte").

Результатом обчислення даного виразу є матриця-стовпець (вектор), кожен елемент якої відповідає розширеному *ASCİİ*-коду відповідного символу (літери) імпортованого до документа повідомлення. У десятковому вигляді коди символів можуть приймати значення від 0 до 255; у двійковому ж для цього є достатнім використання 8 бітів на один символ — так зване однобайтове кодування, на що, зрештою, і вказує параметр "byte" в якості аргументу функції READBİN (див. також додатки Б і Д).

Фрагмент імпортованого повідомлення, а саме коди перших п'ятнадцяти символів (включно з пробілами) у десятковому і двійковому видах, зображено на рис. 5.5[IV].

M =

1	178
2	237
3	244
4	238
5	240
6	236
7	224
8	246
9	179
10	255
11	32
12	186
13	32
14	238
15	228
16	...

M =

1	10110010b
2	11101101b
3	11110100b
4	11101110b
5	11110000b
6	11101100b
7	11100000b
8	11110110b
9	10110011b
10	11111111b
11	100000b
12	10111010b
13	100000b
14	11101110b
15	11100100b
16	...

M_cod =

1	243
2	56
3	113
4	84
5	86
6	80
7	14
8	36
9	237
10	56
11	89
12	251
13	107
14	107
15	74
16	...

Рис. 5.5. Фрагмент повідомлення, що підлягає приховуванню

обрання контейнера належного об'єму (наприклад, для методу заміни НЗБ орієнтовне співвідношення між об'ємами файлів повідомлення і контейнера — 1:8).

IV Для зміни формату відображуваних даних у *Mathcad* необхідно виконати наступне: вибрати операцію *Result* з меню *Format* (або двічі натиснути ліву кнопку маніпулятора безпосередньо на виведених даних) для появи діалогового вікна *Result Format*; на вкладці *Display Options* у випадаючому списку *Radix* обрати необхідну основу системи числення: *decimal* (десяткова), *binary* (двійкова), *octal* (восьмерична) або *hexadecimal* (шістнадцяткова).

Необхідно зауважити, що за умовчанням нижня границя індексації масивів у програмі *Mathcad* дорівнює нулеві. У даній же книзі для зручності індексація починається з одиниці (якщо не буде оговорено іншого), що, зокрема, можна встановити або записом **ORİGİN** := 1 на початку документа, або ж ввести 1 до позиції *Array Origin* на вкладці *Built-In Variables* діалогового вікна *Worksheet Options* з меню *Tools*.

Перевірку імпортування файлу повідомлення (якщо в якості останнього використовується звичайний текст) можна виконати записом **vec2str**(**M**). Зазначена функція повертає рядок символів, що відповідають вектору *ASCII*-кодів **M** [V].

Крок 3

Для стійкості стеганограми до можливих спотворень рекомендується попередньо застосувати один з відомих алгоритмів *завадостійкого кодування*. Ми ж, у свою чергу, перед приховуванням даних текстового файлу "M.txt" у контейнері "C.bmp", захистимо їх *криптографічним кодуванням*, скориставшись простим, але вельми наочним *поліалфавітним шифром Блеза де Віженера (Blaise de Vigenère)*.

Алфавіт джерела повідомлення задаємо у вигляді *ASCII*-кодів: **i** := 1..256; $\mathbf{A}_{\mathbf{i}} := \mathbf{i} - 1$ (для більшої захищеності від зламу, елементи вектора **A** можна переставити за певним законом, що відповідним чином враховувати при розшифровуванні).

Об'єм алфавіту джерела: **Na** := **rows**(**A**), де **rows**(**A**) — функція, що повертає кількість рядків у масиві **A**. У нашому випадку: **Na** = 256 символів.

З символів алфавіту **A** задаємо секретний ключ, наприклад: **K** := "@J|eKc-I98O". Кількість символів у ключі (за відповідною рядковою функцією strlen): **Nk** := **strlen**(**K**) = 11 символів.

Об'єм повідомлення **M**, що підлягає кодуванню: **Nm** := **rows**(**M**) = = 5527 символів (включно зі скритими службовими *ASCII*-символами «переведення рядка» *LF* і «повернення каретки» *CR*, див. додаток І).

[V] У випадку використання символів кирилиці можлива ситуація їх некоректного відображення. Цю проблему можна вирішити декількома способами, описаними, наприклад, за посиланням: http://skesov.ru/kak-ispravit-otobrazhenie-kirillicheskoy-kodirovki-v-windows/ Принагідно зауважимо, що за можливі проблеми в роботі комп'ютера, що можуть виникнути після редагування реєстру чи маніпуляцій з системними файлами, автори даної книги відповідальності не несуть. Крім того, некоректність відображення кириличних символів у системі *Mathcad* жодним чином не впливає на результати стеганографічних перетворень, оскільки алгоритми оперують з двійковими кодами *ASCII*-кодів, а видобуте з контейнера повідомлення можна без проблем прочитати в будь-якому з поширених текстових редакторів.

Розширюємо ключ **K** на довжину повідомлення **M** (**Nm** символів), використовуючи програмний модуль (М.1).

$$\mathbf{K'} := \left|\begin{array}{l} \mathbf{K} \leftarrow \mathbf{str2vec(K)} \\ \mathbf{for}\ \ i \in 1..\mathbf{Nm} \\ \quad \left|\begin{array}{l} r \leftarrow \mathbf{mod}(i, Nk) \\ K'_i \leftarrow \mathbf{if}(r, K_r, K_{Nk}) \end{array}\right. \\ \mathbf{K'} \end{array}\right. \qquad (М.1)$$

У даному модулі функція **str2vec(K)** перетворює рядок символів **K** на вектор їх *ASCII*-кодів. Програмний оператор **for** організовує цикл зміни **i** (змінна циклу) із заданою кількістю повторів (у нашому випадку — від 1 до **Nm** з одиничним кроком приростання). Функція **mod(i,Nk)** повертає залишок від ділення **i** на **Nk**. Якщо залишок від ділення ненульовий, то $\mathbf{K'_i = K_r}$, в іншому випадку — $\mathbf{K'_i = K_{Nk}}$.

Проводимо кодування повідомлення, використовуючи модуль (М.2). *ASCII*-коди перших 15 символів отриманого результату можна побачити на рис. 5.5.

$$\mathbf{M\_cod} := \left|\begin{array}{l} \mathbf{for}\ \ j \in 1..\mathbf{Nm} \\ \quad \left|\begin{array}{l} \mathbf{for}\ \ i \in 1..\mathbf{Na} \\ \quad \left|\begin{array}{l} m \leftarrow i \ \ \mathbf{if}\ \ M_j = A_i \\ n \leftarrow i \ \ \mathbf{if}\ \ K'_j = A_i \end{array}\right. \\ r \leftarrow \mathbf{mod}(m + n, Na) \\ M\_cod_j \leftarrow \mathbf{if}(r, A_r, A_{Na}) \end{array}\right. \\ \mathbf{M\_cod} \end{array}\right. \qquad (М.2)$$

Крок 4

Для того щоб при розпакуванні контейнера з отриманої множини символів можна було чітко визначити початок і кінець саме прихованого повідомлення, доцільно ввести відповідні секретні *мітки*, які б обмежували цей корисний для одержувача зміст.

Аби унеможливити сприйняття за мітки випадкових об'єднань символів у складі прихованого тексту, перші мають складатися з достатньої кількості різних символів. Крім того, для зниження ймовірності виявлення міток при проведенні стеганоаналізу бажано, щоб коди цих символів були достатньо рознесеними на вісі *ASCII* (наприклад, використовувати так звану *транслітерацію*, коли латинські символи перемежовуються символами кирилиці, цифрами, пунктуацією, службовими символами і т. п.; застосовувати псевдовипадкові послідовності символів тощо).

Нехай у нашому випадку обмежувальні мітки матимуть такий вигляд:

μ_s := "n04@m0k"; μ_e := "KiHeu,6".

Дані мітки додаємо до тексту закодованого повідомлення, для чого використовуємо функцію **stack(A, B, ...)**, що дозволяє об'єднувати записані через кому масиви. Об'єднання відбувається шляхом «насадження» матриці **A** на матрицю **B**; отриманої таким чином матриці — на наступну матрицю (якщо така є) і т. д. Зрозуміло, що початкові матриці повинні мати однакову кількість стовпців, тому необхідно перетворити мітки з рядків на вектори *ASCII*-кодів. Таким чином,

$$\mathbf{sMe} := \mathbf{stack}(\mathbf{str2vec}(\mu_s), \mathbf{M\_cod}, \mathbf{str2vec}(\mu_e)).$$

Загальна кількість символів в приховуваному нами повідомленні: **rows(sMe)** = 5541 симв., а кількість НЗБ у складі контейнера, що потрібна для їхнього приховання (8 бітів/символ): 8·**rows(sMe)** = 44328 бітів. Загальна ж кількість доступних НЗБ в обраному нами контейнері становить: **rows(C)**·**cols(C)** = 128·(3·128) = 49152 > 44328 бітів. Таким чином, файл зображення має цілком достатній об'єм для того, щоб приховати у собі повідомлення.

Крок 5

Для подальших обчислень нам буде необхідним переведення десяткового числа (яким за умовчуванням кодується кожен символ) у формат двійкового. Також знадобиться і зворотне перетворення. Оскільки дані функції в *Mathcad* є відсутніми (існує, як це було показано на рис. 5.5, лише можливість перетворення формату відповіді, що для наших цілей не є достатнім), пропонується використати наступні модулі, зміст яких є цілком очевидним.

Перетворення на десяткове двійкового **Q**-розрядного числа **x**, заданого матрицею-стовпцем (причому перший елемент матриці — наймолодший розряд числа), виконується за допомогою модуля (М.3). У свою чергу, перетворення десяткового числа **x** на двійкове **Q**-розрядне реалізується модулем (М.4).

$$\mathbf{B2D}(\mathbf{x}) := \sum_{i=1}^{\mathbf{rows}(\mathbf{x})} \left(\mathbf{x}_i \cdot 2^{i-1}\right) \qquad \text{(М.3)}$$

$$\mathbf{D2B}(\mathbf{x}, \mathbf{Q}) := \left| \begin{array}{l} \textbf{for } i \in 1..\mathbf{Q} \\ \quad \left| \begin{array}{l} V_i \leftarrow \mathbf{mod}(\mathbf{x}, 2) \\ \mathbf{x} \leftarrow \mathbf{trunc}\left(\frac{\mathbf{x}}{2}\right) \end{array} \right. \\ V \end{array} \right. \qquad \text{(М.4)}$$

У (М.4) функція **mod**(**x**, 2) повертає залишок від ділення **x** на 2 («0», якщо **x** парне, і «1», якщо **x** непарне), а функція **trunc**(**z**) — цілу частину дійсного числа **z**.

Крок 6

Для більшої зручності і наочності подальших дій, розгорнемо матрицю **C** у вектор, принагідно змінивши порядок колірних матриць з **R-G-B** на **B-G-R**, що підвищить захищеність прихованої інформації (на даному етапі, звичайно ж, можна скористатися більш надійними, але й складнішими, алгоритмами). У нашому випадку застосуємо модуль (М.5), в якому функція **augment**(**A**,**B**,...) об'єднує матриці **A**, **B**, ... , що мають однакову кількість рядків (об'єднання проводиться стовпець до стовпця, при цьому матриці повинні мати однакову кількість рядків).

$$\begin{aligned} Cv := \;& C' \leftarrow \mathbf{augment}(B, G, R) \\ & \mathbf{for}\ i \in 1..\mathbf{cols}(C') \\ & \quad Cv \leftarrow \mathbf{if}\left(i = 1,\ C'^{\langle i \rangle},\ \mathbf{stack}\left(Cv,\ C'^{\langle i \rangle}\right)\right) \end{aligned} \qquad (M.5)$$

Операція $\mathbf{C}'^{\langle i \rangle}$ дозволяє зчитувати **i**-й стовпець з матриці **C'**, кожен з яких згодом приєднується до результуючого вектора **Cv**. Функція **cols**(**C'**) повертає кількість стовпців у масиві **C'**.

Крок 7

На основі вектора **Cv** формуємо новий вектор, що вже міститиме приховане закодоване повідомлення — модуль (М.6).

$$\begin{aligned} Sv := \;& j \leftarrow 1 \\ & \mathbf{for}\ \mu \in 1..\mathbf{rows}(sMe) \\ & \quad b \leftarrow D2B\left(sMe_{\mu}, 8\right) \\ & \quad \mathbf{for}\ i \in 1..8 \\ & \quad\quad P \leftarrow D2B\left(Cv_j, 8\right),\ P_1 \overset{*}{\leftarrow} b_i \\ & \quad\quad Sv_j \leftarrow B2D(P),\ j \leftarrow j + 1 \\ & \mathbf{while}\ j \le \mathbf{rows}(Cv) \\ & \quad P \leftarrow D2B\left(Cv_j, 8\right),\ P_1 \overset{*}{\leftarrow} \mathbf{round}(\mathbf{rnd}(1)) \\ & \quad Sv_j \leftarrow B2D(P),\ j \leftarrow j + 1 \\ & Sv \end{aligned} \qquad (M.6)$$

Кожен символ (байт) закодованого повідомлення **sMe** (операція циклу **for** $\mu \in 1..$**rows**(**sMe**)) переводиться у двійковий формат (змінна **b**), кожен з восьми розрядів якого записується замість НЗБ (P_1) чисел, що відповідають інтенсивностям того чи іншого кольору восьми обраних

пікселів (останні також попередньо переводяться у двійковий формат (змінна **P**)).

Після виконаної зміни модифіковане двійкове число **P** переводиться у формат десяткового і записується до відповідної позиції вектора **Sv**.

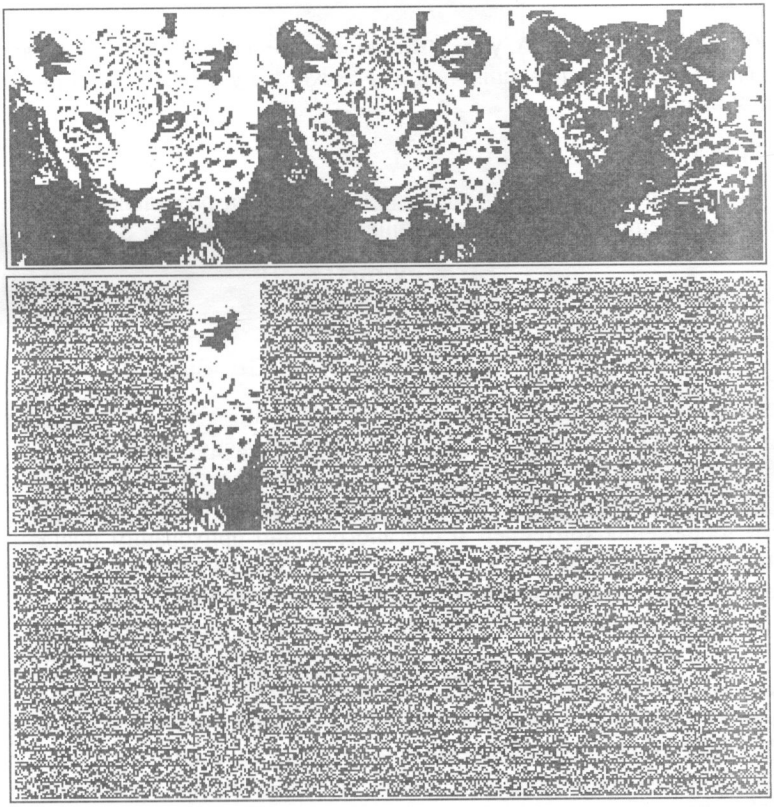
Далі, після обробки останнього символу повідомлення **sMe** проводиться модифікація контейнерів масиву **Cv**, які залишилися пустими. Молодшим бітам кожного з таких елементів присвоюються значення 0 або 1 (у даному випадку — за рівномірним законом розподілу; функція **round**(...) повертає округлене до найближчого цілого значення свого аргументу), хоча більш правильно було б провести дослідження закону розподілу значень вже модифікованих молодших бітів і відповідним чином змінювати ті, що залишилися (дана процедура є темою окремого дослідження і в рамках даної книги не розглядається). Згодом це має значно ускладнити виявити факт модифікованості зображення.

В іншому випадку, проаналізувавши зображення, побудоване з одних лише НЗБ контейнера, порушник у більшості випадків (якщо символів повідомлення «не вистачило» на весь контейнер) виявить границю введення даних і за певних зусиль зможе видобути приховану інформацію. Звичайно, що інформацію ще необхідно буде розшифрувати, але факт її наявності вже буде розкрито і питання захисту повернеться до криптографічної стійкості використаного кодування.

На рис. 5.6 в якості прикладу наведено графічні інтерпретації масивів колірних компонентів, відтворені лише за шарами НЗБ (0/1) контейнера-оригіналу (*а*), контейнера-результату без модифікації (*б*) і з випадковою модифікацією (*в*) надлишкових бітів. Як видно, за відсутності «дописування» при неповному заповненні контейнера чітко простежується границя введення повідомлення (рис. 5.6, *б*) — нагадаємо, що підмасиви **R** і **B** на час заповнення стеганограмою були змінені місцями, тому останньою модифікувалася матриця **R**, яка, зрештою, і виявилася заповненою не до кінця. «Дописування» рівноймовірними 0 і 1 дещо виправляє ситуацію, хоча при додатковому стеганоаналізі на закон розподілу значень НЗБ, а в деяких випадках — навіть неозброєним оком ця невідповідність одразу буде виявлена, що зайвий раз підтверджує необхідність попереднього аналізу розподілу значень вже модифікованих бітів або ж, принаймні, продублювати частину повідомлення для заповнення всього масиву контейнера. Також є очевидною відмінність між рисунками *а* і *в*.

Отже, бажано, щоб зображення, яке планується використати в якості контейнера, було унікальним. Тут треба зауважити, що всі графічні контейнери умовно поділяються на «чисті» і «зашумлені» [95]. У перших простежується зв'язок між молодшим й іншими сімома бітами колірних компонентів, а також залежність між наймолодшими бітами.

Вбудовування повідомлення до «чистого» зображення руйнує існуючі залежності, що, як було показано вище, легко виявляється. Якщо ж зображення вже початково було зашумленим (скановане зображення, цифрова фотографія тощо), то визначення стороннього вкладення стає на порядок важчим, хоча й можливим при використанні теорії ймовірностей і математичної статистики.



а)

б)

в)

Рис. 5.6. Масиви колірних компонентів, відтворені тільки за НЗБ

Також можна зазначити, що для більшої прихованості біти повідомлення слід вносити не послідовно, а лише до кожного 2-го чи навіть 3-го пікселя, або ж узагалі підпорядкувати внесення певному, відомому лише авторизованим особам, закону. Відповідні модифікації легко здійснити шляхом внесення відповідних незначних змін до модуля (M.6).

Крок 8

Отриманий за допомогою модуля (М.6) вектор **Sv** згортаємо в матрицю **S'**, що має розмірність первинної матриці **C** — модуль (М.7):

$$S' := \left| \begin{array}{l} \text{for } i \in 1..\ \text{cols}(C) \\ \quad S'^{\langle i \rangle} \leftarrow \text{submatrix}[Sv, (i-1)\cdot\text{rows}(C)+1, i\cdot\text{rows}(C), 1, 1] \\ S' \end{array} \right. \qquad (M.7)$$

Функція **submatrix**(**A**, **y**, **Y**, **x**, **X**) повертає частину матриці **A**, яка складається з елементів, спільних для рядків від **y** до **Y** та стовпців від **x** до **X** включно.

Крок 9

Виокремлюємо з масиву **S'** колірні матриці та розставляємо їх по належних місцях (**R**↔**B**), одержуючи контейнер-результат **S** (М.8).

$$\begin{aligned} &Bм := \text{extract}(S', 1) \\ &Gм := \text{extract}(S', 2) \\ &Rм := \text{extract}(S', 3) \\ &S := \text{augment}(Rм, Gм, Bм) \end{aligned} \qquad (M.8)$$

Функція **extract**(**M**, **n**) здійснює видобування підмасиву однієї з трьох колірних компонент (**n** = 1, 2 або 3) з загального масиву **M**.

На рис. 5.7 наведено графічну інтерпретацію масиву **S** у вигляді зображення з градаціями сірого та відтворене за колірними складовими зображення-контейнер зі стеганографічно прихованим повідомленням. Порівнюючи рис. 5.7 з рис. 5.3 і 5.4 можна зробити висновок про відсутність будь-яких помітних візуальних відхилень.

Залишається лише записати масив **S** у файл:

$$\text{WRITERGB}(\text{"S\_LSB.bmp"}) := S.$$

Цілком очевидно, що об'єм отриманого файлу відповідатиме об'ємові файлу зображення-оригіналу.

Крок 10

Для дослідження впливу на ступінь прихованості того, до якого з розрядів числа, що характеризує ту або іншу властивість пікселя (у нашому випадку — інтенсивність певного кольору), заноситиметься секретна інформація, у модулі (М.6) у помічених зірочкою рядках слід замість індексу «1» ввести індекс, що відповідатиме модифікованому розряду. Результат, отриманий, наприклад, при внесенні даних до 7-го біта кодових комбінацій інтенсивності кольорів пікселів, зображено на рис. 5.8.

Встановлено, що для більшості зображень, в яких відсутні великі однотонні ділянки, візуально не буде помітним, якщо в якості «носіїв» використовувати не лише наймолодший, але й декілька наступних за ним бітів кожної із зазначених кодових комбінацій.

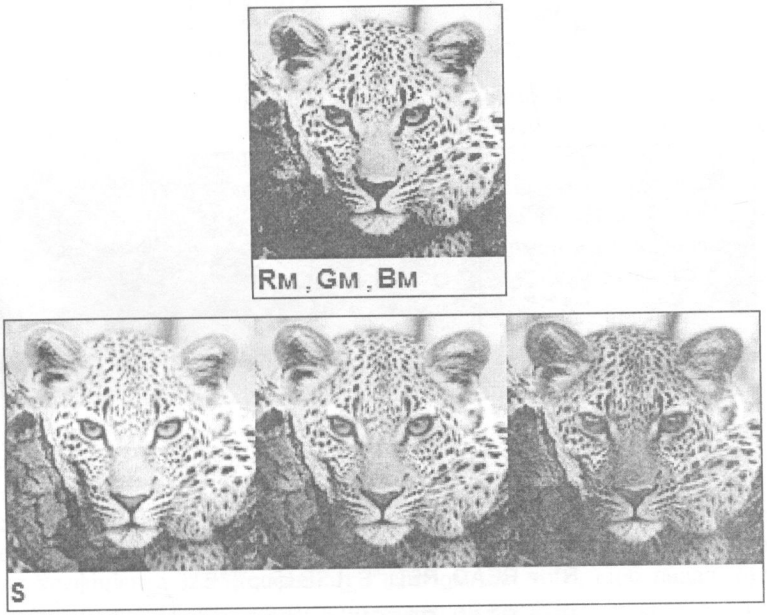


Рис. 5.7. Контейнер-результат та його інтерпретація у вигляді масиву колірних компонентів

Таким чином, в якості одного з можливих ступенів додаткового захисту бачиться використання змінюваного за певним законом почергового запису до цих бітів у різних пікселях. Або ж, жертвуючи прихованістю, можна вдвічі чи втричі збільшити пропускну здатність створюваного стеганоканалу.

Крок 11

Розглянемо процес видобування прихованого повідомлення. Попередньо знаючи, що повідомлення було поміщене до масиву колірних компонентів, виділяємо відповідні кожному кольору підмасиви, переводячи значення колірних характеристик кожного пікселя зображення, що містить у собі закодовану стеганограму, у числові матриці:

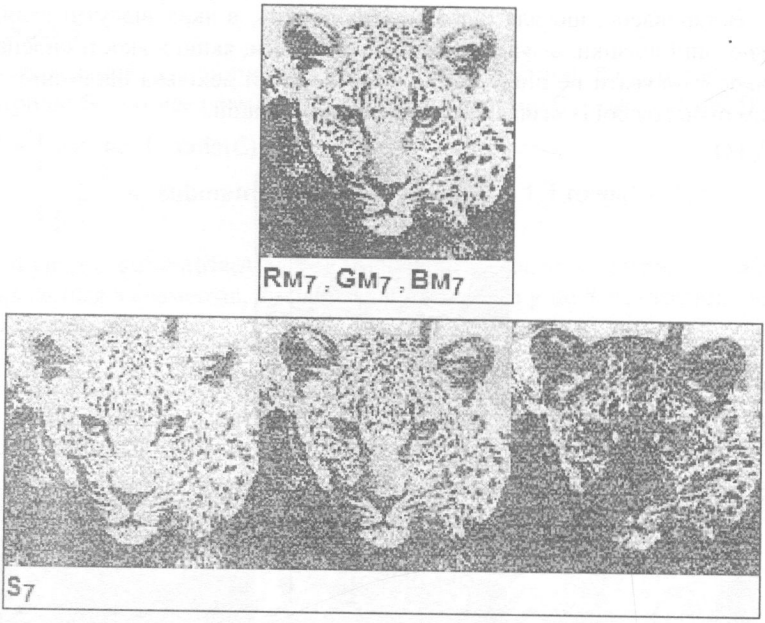


Рис. 5.8. Кольорове зображення та масив колірних складових у випадку внесення приховуваних даних до 7-го біта інтенсивностей кольорів

$$R^* := \mathbf{READ\_RED}(\text{"S\_LSB.bmp"})^{VI};$$
$$G^* := \mathbf{READ\_GREEN}(\text{"S\_LSB.bmp"});$$
$$B^* := \mathbf{READ\_BLUE}(\text{"S\_LSB.bmp"}).$$

З отриманих матриць, відповідним чином змінюючи їх порядок (подібно до (М.5)), формуємо вектор **Sv*** — модуль (М.9):

$$Sv^* := \left| \begin{array}{l} S' \leftarrow \mathbf{augment}(B^*, G^*, R^*) \\ \mathbf{for}\ \ i \in 1 .. \mathbf{cols}(S') \\ \quad Sv^* \leftarrow \mathbf{if}\left(i = 1, S'^{\langle 1 \rangle}, \mathbf{stack}\left(Sv^*, S'^{\langle i \rangle}\right)\right) \end{array} \right. \qquad (М.9)$$

Належним чином обробляючи кожні вісім елементів отриманого вектора, використовуючи модуль (М.10), видобуваємо приховане повідомлення.

[VI] Тут і надалі символом «*» позначатимуться параметри, що відносяться до приймальної сторони. Також за умовчанням вважатиметься, що приймальна сторона обізнана з алгоритмом вбудовування.

$$\mathbf{Mf^*} := \left| \begin{array}{l} \text{for } \mu \in 1..\ \dfrac{\text{rows}(\mathbf{Sv^*})}{8} \\ \quad \left| \begin{array}{l} \text{for } i \in 1..\ 8 \\ \quad b_i \leftarrow \text{D2B}\left[\mathbf{Sv^*}_{i+8\cdot(\mu-1)},\ 8\right]_1 \\ \mathbf{Mf^*}_\mu \leftarrow \text{B2D}(b) \\ \mathbf{Mf^*}_\mu \leftarrow \mathbf{Mf^*}_\mu + 32.5 \quad \text{if } \mathbf{Mf^*}_\mu < 32 \end{array} \right. \\ \mathbf{Mf^*} \end{array} \right. \qquad \text{(M.10)}$$

Слід зазначити, що оскільки наперед невідомо, яку частину вектора **Sv*** займає саме корисна інформація (стеганограма), до уваги беруться всі його елементи. Значення кожного елемента формованого при цьому вектора **Mf*** являють собою коди символів «квазіповідомлення», що обчислюються у зворотному до (М.6) порядку: кожен молодший розряд вісімки перетворених у двійковий формат елементів вектора **Sv*** формує двійкове число *ASCII*-коду символу, формат якого згодом перетворюється на десятковий. Отримане число присвоюється μ-му елементові вектора **Mf***.

Крок 12

У зв'язку з неможливістю оброблення рядковими функціями останніх версій *Mathcad* (починаючи з 12-ї) символів, *ASCII*-коди яких мають значення від 0 до 31 включно (за винятком службових символів *LF* (код 10) і *CR* (код 13)), додатково виконується «маркування» елементів вектора **Mf***, що мають значення 0, 1, 2, ... , 31, додаванням до кожного з них коефіцієнта 32.5 (коефіцієнт є дробом для того, щоб надалі було можливим відрізнити «справжні» значення елементів масиву від тих, що перед цим мали неформатні для *Mathcad* значення). Таку заміну, звичайно ж, відповідним чином необхідно буде врахувати у подальшому, для чого запам'ятаємо номери рядків вектора **Mf***, елементи яких мають дробові значення — модуль (М.11).

$$\mathbf{N} := \left| \begin{array}{l} \text{for } s \in 0..\ 31 \\ \quad \left| \begin{array}{l} i \leftarrow 1 \\ \text{for } \mu \in 1..\ \text{rows}(\mathbf{Mf^*}) \\ \quad \text{if } \mathbf{Mf^*}_\mu = s + 32.5 \\ \quad \quad \left| \mathbf{N}_{i,\,s+1} \leftarrow \mu\ ,\ i \leftarrow i+1 \right. \end{array} \right. \\ \mathbf{N} \end{array} \right. \qquad \text{(M.11)}$$

При цьому, до 1-го стовпця формованого масиву **N** заносяться номери елементів, значення яких дорівнюють 32.5 (колишні коди нулів), до 2-го стовпця — номери елементів зі значенням 33.5 (колишні коди одиниць), і т. д., до стовпця № 32 (колишні коди 31).

Крок 13

Знаючи, що текст корисної інформації був обмежений мітками μ^*_s := "n04@m0k" і μ^*_e := "KiHeu,6", виокремлюємо його з видобутого квазіповідомлення, використовуючи модуль (М.12).

$$
\begin{array}{l}
\mathbf{M\_cod^*} := \left|
\begin{array}{l}
\mathbf{s} \leftarrow 0\,,\ \mathbf{e} \leftarrow 0 \\
\beta_s \leftarrow \mathbf{strlen}(\mu_s)\,,\ \beta_e \leftarrow \mathbf{strlen}(\mu_e) \\
\mathbf{Mf^*} \leftarrow \mathbf{vec2str}(\mathbf{Mf^*}) \\
\mathbf{for}\ \ \mu \in 1\,..\,\mathbf{strlen}(\mathbf{Mf^*}) \\
\quad \left|
\begin{array}{l}
\mathbf{s} \leftarrow \mu + \beta_s \ \ \mathbf{if}\ \ \mathbf{substr}(\mathbf{Mf^*}, \mu, \beta_s) = \mu_s \wedge \mathbf{s} = 0 \\
\mathbf{e} \leftarrow \mu - 1 \ \ \mathbf{if}\ \ \mathbf{substr}(\mathbf{Mf^*}, \mu, \beta_e) = \mu_e \wedge \mathbf{e} = 0 \\
\mathbf{break}\ \ \mathbf{if}\ \ \mathbf{s} \neq 0 \wedge \mathbf{e} \neq 0
\end{array}
\right. \\
\mathbf{M\_cod^*} \leftarrow \mathbf{str2vec}(\mathbf{substr}(\mathbf{Mf^*}, \mathbf{s}, \mathbf{e} - \beta_s)) \\
\mathbf{for}\ \ \mathbf{n} \in 1\,..\,\mathbf{cols}(\mathbf{N}) \\
\quad \mathbf{for}\ \ \mathbf{i} \in 1\,..\,\mathbf{rows}(\mathbf{N}) \\
\quad\quad \left|
\begin{array}{l}
\mathbf{break}\ \ \mathbf{if}\ \ \mathbf{N}_{\mathbf{i},\mathbf{n}} = 0 \\
\mathbf{M\_cod^*}_{\mathbf{N}_{\mathbf{i},\mathbf{n}} - \beta_s} \leftarrow \mathbf{n} - 1 \ \ \mathbf{if}\ \ 0 < \mathbf{N}_{\mathbf{i},\mathbf{n}} \leq \mathbf{rows}(\mathbf{M\_cod^*}) + \beta_s
\end{array}
\right. \\
\mathbf{M\_cod^*}
\end{array}
\right.
\end{array}
\qquad (\text{М.12})
$$

$\mu - 1$

$\mathbf{s} - 1$

Вектор *ASCII*-кодів **Mf***, попередньо перетворений за допомогою функції **vec2str**(...) на відповідний йому рядок символів, послідовно проходиться у пошуку стартової та кінцевої міток. Така операція виконується шляхом порівняння виокремленої частини рядку даних з відповідними мітками, які повинні бути відомими одержувачеві.

Саме ж виокремлення проводиться за допомогою функції **substr**(**Mf***, μ, β), що повертає підрядок довжиною у β символів з рядка **Mf***, починаючи з символу μ (треба зауважити, що в даному випадку перший символ рядка має номер, що відповідає значенню вбудованої змінної **ORIGIN**)[VII]. У нашому випадку, оскільки кожна з міток складається з семи символів, β = 7.

Просування вздовж рядка даних **Mf*** відбувається послідовним збільшенням μ. При виконанні зазначених у (М.12) умов (оператори **if**), коефіцієнтам **s** і **e** присвоюються відповідні значення номерів початкової і кінцевої позицій корисної інформації в рядку даних **Mf***.

[VII] У меню *Tools* ► *Worksheet Options*, на вкладці *Calculation* слід встановити прапорець навпроти принципу *Use ORIGIN for string indexing*. Дана можливість є відсутньою у старих версіях *Mathcad*, в яких перший символ рядку завжди мав індекс 0. У цих версіях для коректної роботи модуля (М.12) необхідно виконати зміни μ на μ – 1 та **s** на **s** – 1 у зазначених на рисунку позиціях.

Такі додаткові умови, як $\mathbf{s} = 0$; $\mathbf{e} = 0$; $\mathbf{s} \neq 0$; $\mathbf{e} \neq 0$, введені для пришвидшення пошуку.

Зворотне перетворення рядка символів **Mf*** на вектор їх *ASCII*-кодів дозволяє в цьому ж модулі провести безпосереднє виокремлення прихованої інформації та відновити елементи, значення яких були примусово змінені на дробові.

Крок 14

Видобуте повідомлення треба декодувати. Необхідні початкові умови на цьому етапі — авторизованій стороні є відомими:

- алфавіт джерела повідомлення ($\mathbf{i} := 1..256$; $\mathbf{A^*_i} := \mathbf{i} - 1$) об'ємом $\mathbf{Na^*} := \mathbf{rows}(\mathbf{A^*}) = 256$ символів;
- секретний ключ **K*** := "@J|eKc-I98O" з $\mathbf{Nk^*} := \mathbf{strlen}(\mathbf{K^*}) = 11$ симв.;
- довжина повідомлення: $\mathbf{Nm^*} := \mathbf{rows}(\mathbf{M\_cod^*}) = 5527$ символів.

Секретний ключ **K*** розширюється на довжину **Nm*** повідомлення **M_cod*** аналогічно тому, як це робилося при кодуванні повідомлення — див. (М.1). Власне ж декодування секретного повідомлення проводиться за допомогою модуля (М.13).

$$\begin{array}{l}\mathbf{M^*} := \left|\begin{array}{l}\textbf{for}\ \ \mathbf{j} \in 1..\mathbf{Nm^*} \\ \quad \left|\begin{array}{l}\textbf{for}\ \ \mathbf{i} \in 1..\mathbf{Na^*} \\ \quad\quad \left|\begin{array}{l}\mathbf{m} \leftarrow \mathbf{i}\ \ \textbf{if}\ \ \mathbf{M\_cod^*_j} = \mathbf{A^*_i} \\ \mathbf{n} \leftarrow \mathbf{i}\ \ \textbf{if}\ \ \mathbf{K'^*_j} = \mathbf{A^*_i}\end{array}\right. \\ \mathbf{r} \leftarrow \mathbf{mod}(\mathbf{Na^*} + \mathbf{m} - \mathbf{n}, \mathbf{Na^*}) \\ \mathbf{M^*_j} \leftarrow \textbf{if}(\mathbf{r}, \mathbf{A^*_r}, \mathbf{A^*_{Na^*}})\end{array}\right. \\ \mathbf{M^*}\end{array}\right. \qquad \text{(М.13)}$$

Декодоване повідомлення **M*** записуємо у текстовий файл:

$$\mathbf{WRITEBIN}(\text{"M\_dec.txt"}, \text{"byte"}, 0) := \mathbf{M^*}.$$

Крок 15

Проведемо обчислення показників візуального спотворення, наведених у розділі 3 (формули (3.1)–(3.17)). Отримані результати зведемо до табл. 5.6: *«а»* — стеганограмою є частина вступу до даної книги, вбудована до масиву **B**; *«б»* — повідомлення є рядком "© Alex Puzyrenko, 2017" (див. стор. 164).

5.3.2.2. Метод псевдовипадкового інтервалу

У вищерозглянутому найпростішому випадку проводиться заміна НЗБ усіх послідовно розташованих пікселів зображення. Інший підхід

— *метод випадкового інтервалу* [78], полягає у випадковому розподілі бітів секретного повідомлення по контейнеру, у результаті чого відстань між двома вбудованими бітами визначається псевдовипадково. Ця методика особливо ефективна у випадку, коли бітова довжина секретного повідомлення є істотно меншою за кількість пікселів зображення.

Розглянемо найпростіший випадок даного методу, коли інтервал між двома послідовними вбудовуваннями бітів повідомлення є функцією координат попередньо модифікованого пікселя.

Крок 1

Нехай повідомлення, яке необхідно приховати: **M** := "© Alex Puzyrenko, 2017". У якості контейнера **C** використаємо підмасив **B** синьої колірної компоненти зображення рис. 5.3.

Крок 2

Внесемо мітки, що визначатимуть границі корисного повідомлення у контейнері. На відміну від попереднього методу, стартова мітка визначатиме порядковий номер елемента контейнера, починаючи з якого в останній заноситимуться дані. Нехай $\mu_s := 107$. Мітка μ_e сигналізуватиме про завершення корисної частини серед видобутих символів. Нехай, як і раніше, μ_e := "KiHeu,6".

Крок 3

Приймемо, що для внесення бітів повідомлення до контейнера зі змінним кроком, величина останнього обумовлюється кількістю одиниць (вагою) у двійковому еквіваленті номера елемента контейнера, який модифікувався попередньо. Для підрахунку величини кроку (інтервалу) в такий спосіб, скористаємося модулем (М.14), що підсумовує кількість одиничних символів матриці-стовпця **x**.

$$\mathbf{step}(\mathbf{x}) := \mathbf{round}\left(\mathbf{K}\cdot\sum_{i=1}^{\mathbf{rows}(\mathbf{x})}\mathbf{x}_i\right). \qquad (М.14)$$

Коефіцієнт **K** у даному випадку виступає у ролі найпростішого ключа і може приймати будь-які дійсні значення (у тому числі й від'ємні, але у цьому випадку стартова мітка μ_s повинна отримати значення, близьке до найбільшого значення індексу елементів контейнера). Також при обранні **K** слід брати до уваги загальну кількість бітів, необхідну для приховання повідомлення, а також наявну кількість елементів у масиві контейнера. Нехай **K** := 9,5.

За допомогою рядкової функції **concat**(...), яка об'єднує рядки, що є її аргументами, додамо до тексту повідомлення, що підлягає приховуванню, обмежуючу мітку μ_e. Результат об'єднання перетворимо на вектор *ASCII*-кодів: **Me** := **str2vec**(**concat**(**M**, μ_e)).

Загальна кількість символів в одержаному повідомленні становить **rows**(**Me**) = 29. Кількість НЗБ контейнера, що будуть потрібні для їх розміщення (8 бітів/символ): 8·**rows**(**Me**) = 232 біти.

Крок 4

Розгорнемо масив **B** у вектор **Cv** — (M.15), на основі якого сформуємо новий вектор **Sv**, що міститиме приховане повідомлення — (M.16).

$$\mathbf{Cv} := \mathbf{for}\ \ i \in 1..\mathbf{cols}(\mathbf{B}) \qquad \text{(M.15)}$$
$$\mathbf{Cv} \leftarrow \mathbf{if}\left[i = 1, \mathbf{Cv} \leftarrow \mathbf{B}^{\langle 1 \rangle}, \mathbf{stack}\left[\mathbf{Cv}, \mathbf{B}^{\langle i \rangle}\right]\right]$$

$$\mathbf{Sv} := \left|\begin{array}{l} \mathbf{Sv} \leftarrow \mathbf{Cv} \\ z \leftarrow \mu_s \\ \mathbf{for}\ \ \mu \in 1..\mathbf{rows}(\mathbf{Me}) \\ \quad \left|\begin{array}{l} \mathbf{b} \leftarrow \mathbf{D2B}(\mathbf{Me}_\mu, 8) \\ \mathbf{for}\ \ i \in 1..8 \\ \quad \left|\begin{array}{l} z \leftarrow z + \mathbf{step}(\mathbf{D2B}(z, \mathbf{ceil}(\log(z, 2)))) \\ \mathbf{P} \leftarrow \mathbf{D2B}(\mathbf{Cv}_z, 8),\ \mathbf{P}_1 \leftarrow \mathbf{b}_i \\ \mathbf{Sv}_z \leftarrow \mathbf{B2D}(\mathbf{P}) \end{array}\right. \end{array}\right. \\ \mathbf{Sv} \end{array}\right. \qquad \text{(M.16)}$$

Кожен символ повідомлення з обмежувальною міткою **Me** (операція циклу **for** $\mu \in$ 1..**rows**(**Me**)) переводиться у двійковий формат (змінна **b**), кожен розряд якого записується замість наймолодших бітів чисел **P**, що відповідають значенням інтенсивності синього кольору певної вісімки пікселів. При цьому елементи масиву **Cv** проходяться не послідовно, а зі змінним кроком, величина якого обумовлюється функцією **step**(...)[VIII].

Стартовий елемент задається міткою μ_s. Після проведеної зміни НЗБ модифіковане двійкове число **P** переводиться у формат десяткового і записується у відповідну позицію вектора **Sv**, який на початку модуля був прийнятий рівним вектору **Cv**.

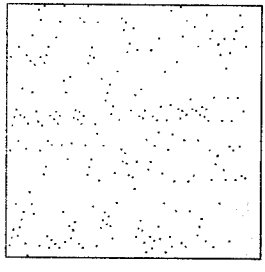
[VIII] При обчисленні функції **D2B**(**z**, **Q**) визначається достатня кількість розрядів для представлення десяткового числа у двійковому форматі. Функція **ceil**(**x**) повертає найменше ціле число, таке, що перевищує або ж дорівнює **x**.

Крок 5

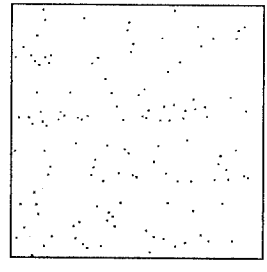
Зворотне згортання вектора **Sv** до масиву, що має розмірність контейнера-оригіналу, здійснюється з використанням модуля (М.7) з тією лише відмінністю, що цього разу аргументом функцій розмірності масиву **rows**(...) і **cols**(...) є масив **B**.

Щоб оцінити ступінь «розсіювання» бітів прихованого повідомлення по масиву контейнера, в якості прикладу наведемо результат присвоєння пікселю, до якого планувалося ввести біт повідомлення, нульового значення інтенсивності (чорний колір) при попередньому загальному висвітленні зображення ((М.17), рис. 5.9, *а*). Пікселі контейнера, яскравість яких зазнала змін за результатом вбудовування повідомлення, показані на рис. 5.9, *б* (усього — 125 пікселів).

$$Sv\_ := \left| \begin{array}{l} Sv \leftarrow \dfrac{Cv + 345}{\max(Cv) + 345} \cdot 255 \\ z \leftarrow \mu_S \\ \text{for } \mu \in 1..8 \cdot rows(Me) \\ \quad \left| \begin{array}{l} z \leftarrow z + step(D2B(z, ceil(\log(z, 2)))) \\ Sv_z \leftarrow 0 \end{array} \right. \\ Sv \end{array} \right. \qquad \text{(М.17)}$$



а)



б)

Рис. 5.9. Приклад розсіювання бітів повідомлення по масиву контейнера

Крок 6

Результуюче кольорове зображення визначатиметься масивом об'єднання колірних масивів: **S := augment(R, G, S')**.

Крок 7

При видобуванні прихованого повідомлення, одержувачеві мають бути відомими параметри μ^*_S, μ^*_e, K^* і, зрештою, номер каналу колірності, до якого здійснювалося вбудовування стеганограми.

Розгортання масиву **B*** у вектор **Sv*** відбувається за аналогічним до (М.15) модулем. Видобування повідомлення з вектора **Sv*** проводиться за допомогою модуля (М.18) у порядку, зворотному операції вбудовування.

З одержаного вектора **Mf*** шляхом порівняння з міткою μ^*_e виділеного фрагмента видобувається корисне повідомлення **M*** (М.19).

```
Mf* := | z ← μ*_s                                                    (M.18)
       | for μ ∈ 1.. rows(Sv*)
       |   | for i ∈ 1.. 8
       |   |   | z ← z + step(D2B(z, ceil(log(z, 2))))
       |   |   | break if z > rows(Sv*)
       |   |   | P ← D2B(Sv*_z, 8)
       |   |   | b_i ← P_1
       |   | Mf*_μ ← B2D(b)
       | Mf*
```

```
M* := | μ*_e ← str2vec(μ*_e), β_e ← rows(μ*_e)                      (M.19)
      | for μ ∈ 1.. rows(Mf*) − β_e
      |   if submatrix(Mf*, μ, μ+β_e−1, 1, 1) = μ*_e
      |     M* ← submatrix(Mf*, 1, μ−1, 1, 1)
      | vec2str(M*)
```

У підсумку маємо:

M* = "© Alex Puzyrenko, 2017".

Крок 8

Результати обчислення візуального спотворення контейнера зведені до табл. 5.6 (див. стор. 164).

5.3.2.3. Метод псевдовипадкової перестановки

Недоліком методу псевдовипадкового інтервалу є те, що біти повідомлення розміщуються у контейнері у тій самій послідовності, що й в самому повідомлені, і лише інтервал між ними змінюється псевдовипадковим чином. Тому для контейнерів фіксованого розміру більш оптимальним є використання методу *псевдовипадкової перестановки (обрання)* [79], сутність якого полягає у тому, що k-й біт повідомлення зберігається у пікселі з індексом j_k, а останній є елементом з послідовності $j_1, j_2, \ldots, j_M$, попередньо створеної генератором ПВЧ.

Нехай N — загальна кількість бітів (наймолодших) у наявному контейнері; P^N — перестановка чисел $\{1, 2, \ldots, N\}$. Тоді, якщо ми маємо для приховання конфіденційне повідомлення довжиною n бітів, можна просто вбудувати ці біти замість бітів контейнера $P^N(1)$, $P^N(2)$, ..., $P^N(n)$.

Функція перестановки повинна бути псевдовипадковою, тобто вона має забезпечувати обрання бітів контейнера приблизно випадковим чином. Тільки у цьому випадку секретні біти будуть рівномірно розподілені по всьому бітовому про́стору контейнера. Утім, обов'язковою умовою є те, що індекс певного біта контейнера може з'явитися у послідовності не більше одного разу, іншими словами, не повинне відбуватися «перетинання», що, певна річ, призводитиме до спотворення вже вбудованого біта.

Якщо кількість бітів повідомлення є набагато меншою за кількість НЗБ зображення, то ймовірність такого перетинання є незначною, і спотворені біти можуть бути відновлені за допомогою коригувальних кодів. Імовірність, принаймні, одного перетинання оцінюється як:

$$p \approx 1 - exp\left[-\frac{l_M \cdot (l_M - 1)}{2 \cdot l_C}\right], \quad l_M << l_C. \qquad (5.1)$$

При $l_C = const$ і збільшенні l_M дана ймовірність прямує до одиниці.

Крім того, для запобігання перетинань можна запам'ятовувати всі індекси використаних елементів j_i, а перед модифікацією кожного нового пікселя проводити перевірку його координат на повторюваність. Також можна застосовувати генератори ПВЧ без повторюваності чисел. Останній випадок розглянемо більш докладно.

Для наших цілей функція перестановки також залежить від секретного ключа K. При цьому генератор псевдовипадкової перестановки P^N є функцією, що для кожного значення K виробляє різні псевдовипадкові перестановки чисел $\{1, 2, \ldots, N\}$.

Позначимо через P_K^N генератор перестановок з відповідним ключем K. Якщо перестановка P_K^N є захищеною по обчисленню (тобто, розкриття вимагає від порушника невиправдано великих витрат обчислювальних ресурсів), то можливість розкриття змісту або припущення самого лише виду перестановок без володіння інформацією про секретний ключ K практично дорівнює нулеві.

Секретний генератор псевдовипадкової перестановки (ГПВП) може бути ефективно реалізований на основі генератора псевдовипадкової функції (ГПВФ) [98], котрий, як і ГПВП, для кожного окремого значення ключа виробляє різні функції, що не піддаються прогнозуванню;

проте, множина значень функції не повинна дорівнювати області її визначення. ГПВФ легко реалізується з секретної хеш-функції H шляхом об'єднання аргументу i з секретним ключем K та взяття від результуючого бітового рядка функції H:

$$f_K(i) = H(K \circ i), \tag{5.2}$$

де $K \circ i$ — об'єднання (конкатенація) бітових рядків K та i; $f_K(i)$ — результуюча псевдовипадкова функція від i, що залежить від параметра K.

Генератор Майкла Лубі *(M. Luby)* та Чарлза Рекоффа *(C. Reckoff)* побудований наступним чином. Нехай i — рядок двійкових даних довжиною $2 \cdot l$. Розділимо i на дві частини: x та y довжиною l кожна, а ключ K — на чотири частини: K_1, K_2, K_3 і K_4. Тоді

$$y = y \oplus f_{K_1}(x) = y \oplus H(K_1 \circ x);$$

$$x = x \oplus f_{K_2}(y) = x \oplus H(K_2 \circ y);$$

$$y = y \oplus f_{K_3}(x) = y \oplus H(K_3 \circ x);$$

$$x = x \oplus f_{K_4}(y) = x \oplus H(K_4 \circ y);$$

повернення $y \circ x$.

Під записом виду $a \oplus b$ розуміється побітове додавання за модулем 2 аргументу a до аргументу b, причому результат додавання має ту саму розмірність, що й a.

Для кожного значення ключа K алгоритм повертає псевдовипадкову перестановку з чисел $\{1, \ldots, 2^{2l}\}$. Лубі та Рекофф показали, що перестановка є настільки ж секретною, наскільки й генератор ПВФ. Вони також навели простий алгоритм перестановки з $\{1, \ldots, 2^{2l+1}\}$. Якщо значення функції f_K являють собою досить довгі бітові послідовності, той самий ефект можна отримати, прийнявши, що y — це перші l бітів рядка i, а x — останні $(l+1)$ бітів.

Вищенаведена конструкція дозволяє отримати перестановку $P_K^{2^k}$ з $\{1, \ldots, 2^k\}$ для довільного k. Проте, у випадку, коли кількість бітів контейнера становить N, виникає необхідність у перестановці P_K^N з $\{1, \ldots, N\}$.

Перевагою запропонованого у [79] методу є те, що з'являється можливість обмежитися лише наявними для P_K^N аргументами. Нехай $k = \lceil log_2(N) \rceil$ (прямокутні дужки означають округлення до найменшого цілого, що більше або дорівнює аргументу). Тоді $2^{k-1} < N \le 2^k$. При

цьому підраховуються значення $P_K^{2^k}(1)$, $P_K^{2^k}(2)$, ... і з послідовності видаляються будь-які числа, що перевищують N. Таким чином одержують значення $P_K^N(1)$, $P_K^N(2)$, ... Зауважимо, що це стає можливим, коли функція перестановки обчислена для зростаючих значень аргументів, починаючи з одиниці. Отже, алгоритм Лубі та Рекоффа цілком придатний для побудови генератора ПВП P^N для довільного N.

Однак, коли N є складеним (а у випадку зображення так і є), існує більш зручний спосіб побудови ГПВП. Наведений нижче алгоритм базується на блочному кодуванні з довільним розміром блока [79, 99].

Кількість бітів контейнера повинна становити собою складене число з двох співмножників приблизно однакового порядку, тобто $N = X \cdot Y$ для деяких X та Y. У випадку, коли дані приховуються у НЗБ пікселів цифрового зображення, параметри X та Y є розмірами даного зображення. Для одержання координат i-го пікселя зображення для приховання у ньому біта повідомлення ($i \in \{1, \ldots, N\}$) необхідно виконати наступні обчислення:

$$x = div(i, Y) + 1; \qquad (5.3\ а)$$

$$y = mod(i, Y) + 1; \qquad (5.3\ б)$$

$$x = mod\left(x + f_{K_1}(y), X\right) + 1; \qquad (5.3\ в)$$

$$y = mod\left(y + f_{K_2}(x), Y\right) + 1; \qquad (5.3\ г)$$

$$x = mod\left(x + f_{K_3}(y), X\right) + 1; \qquad (5.3\ д)$$

$$i = (x, y) \text{ або } i = (x - 1) \cdot Y + y, \qquad (5.3\ е)$$

де $div(i, X)$ і $mod(i, X)$ — функції, що повертають, відповідно, ціле і залишок від ділення i на X. Другий варіант формули (5.3 *е*) застосовний у випадку, якщо масив зображення попередньо було розгорнуто у вектор (по рядках). Додавання одиниці необхідне у випадку індексації елементів масиву зображення, починаючи з 1.

Перші два раунди алгоритму ((5.3 *а*) – (5.3 *г*)) необхідні для того, щоб «розосередити» біти приховуваного повідомлення серед найменш значущих бітів контейнера. При цьому перший раунд надає випадкового характеру x-координатам пікселя-контейнера, а другий — y-координатам. Третій раунд необхідний для запобігання атаці на відкритий (незашифрований) текст.

У випадку використання лише двох раундів, наприклад, нехай $i = (b - 1) \cdot Y + a$, а $P_K^N(i)$ — значення перестановки. Якщо криптоаналітик здатен припустити значення a і для деякого z може одержати

пару «відкритий текст — закодований текст»: $i' = (z-1)\cdot Y + a$; $P_K^N(i')$, то він здатний також і встановити b. Навіть при тому, що у [79] висловлюється припущення, що запропонований алгоритм буде достатньо стійким і з трьома раундами, також визнається, що у деяких випадках може знадобитися чотири і більше раундів, що ще більше підвищить стійкість алгоритму до зламу.

Промоделюємо даний метод у програмі *Mathcad*.

Крок 1

Повідомлення, яке треба приховати: **M** := "© Alex Puzyrenko, 2017". Контейнер **C** — підмасив **B** синьої колірної компоненти зображення рис. 5.3. При цьому кількість бітів у повідомленні: L_M := **strlen(M)** = = 176 бітів; геометричні розміри контейнера: **X** := **rows(C)** = 128 пікс.; **Y** := **cols(C)** = 128 пікс.; загальна кількість доступних для модифікації НЗБ: **N** := **X·Y** = 16384 бітів.

Крок 2

Для формування ключа використаємо програмний модуль (М.20), що дозволяє на основі первинного ключа $K_0 \geq 2$ сформувати вектор, що міститиме $\Re$ пар ключів (кожна пара ключів використовуватиметься у відповідному раунді обчислення координат **x** та **y**).

$$K := \left| \begin{array}{l} K_1 \leftarrow K_0 \\ \text{for } s \in 2..2\cdot\Re \\ \quad K_s \leftarrow \text{str2num}\left[\text{substr}\left[\text{num2str}\left[\left(K_{s-1}\right)^2\right], 1, 3\right]\right] \\ \quad K_s \leftarrow \text{str2num}\left[\text{substr}\left[\text{num2str}\left[K_s\right], 1, 2\right]\right] \text{ if } K_s > 255 \\ K \end{array} \right. \qquad \text{(М.20)}$$

0

У даному модулі почергово використовується три функції:

- **num2str(d)** — перетворення аргументу-числа **d** на відповідний рядок **A**;
- **substr(A, 1, 3)** — виділення з рядка **A** фрагмента, що складається з трьох його перших символів[IX];
- **str2num(a)** — зворотне перетворення рядкового фрагмента **a** на число, що, зрештою, і присвоюється **s**-му елементові масиву **K**.

У випадку отримання $K_s > 255$, за допомогою аналогічної комбінації функцій число скорочується на одну цифру.

[IX] Див. виноску № VII.

На рис. 5.10 наведено приклад обчислення пар ключів при $K_o = 123$ і $\Re = 7$.

$K^T =$

1	2	3	4	5	6	7	8	9	10	11	12	13	14
123	15	228	5	26	67	44	193	37	136	184	33	108	116

Рис. 5.10. Приклад результату обчислення ключів **K**

Крок 3

Вбудовування бітів повідомлення до псевдовипадкових пікселів контейнера виконаємо за модулем (М.21), що реалізує алгоритм (5.3). На початку модуля масив **S** прирівнюється до первинного масиву **C**. Також проводиться конвертування повідомлення з рядкового формату на вектор двійкових даних M_{vec_bin}. При обчисленні координат **x** та **y** використовується операція векторизації (*вираз*), яка у даному випадку дозволяє послементно додавати за модулем 2 двійкові вектори **K** та **y** (або **x**).

$$
\begin{aligned}
S := \Big| \; & S \leftarrow C \\
& M_{vec} \leftarrow \text{str2vec}(M) \,;\, M_{vec\_bin} \leftarrow \text{D2B}\left(M_{vec_1}, 8\right) \\
& \text{for } j \in 2..\text{rows}\left(M_{vec}\right) \\
& \quad M_{vec\_bin} \leftarrow \text{stack}\left(M_{vec\_bin}, \text{D2B}\left(M_{vec_j}, 8\right)\right) \\
& \text{for } i \in 1..L_M \\
& \quad x \leftarrow \text{trunc}\left(\frac{i}{Y}\right) + 1 \,,\; y \leftarrow \text{mod}(i, Y) + 1 \\
& \quad \text{for } s \in 1..\Re \\
& \quad\quad x \leftarrow \text{mod}\left[x + \text{B2D}\left[\overrightarrow{\left(\text{D2B}\left(K_{2\cdot s-1}, 8\right) \oplus \text{D2B}(y, 8)\right)}, X\right] + 1\right] \\
& \quad\quad y \leftarrow \text{mod}\left[y + \text{B2D}\left[\overrightarrow{\left(\text{D2B}\left(K_{2\cdot s}, 8\right) \oplus \text{D2B}(x, 8)\right)}, Y\right] + 1\right] \\
& \quad P \leftarrow \text{D2B}\left(C_{x,y}, 8\right) \,;\, P_1 \leftarrow M_{vec\_bin_i} \,;\, S_{x,y} \leftarrow \text{B2D}(P) \\
& S
\end{aligned}
\qquad \text{(M.21)}
$$

Ступінь «розсіювання» бітів повідомлення по масиву контейнера при різній кількості раундів $\Re$ обчислення координат **x** та **y** можна оцінити за рис. 5.11 (формування рисунку здійснено за тим самим

принципом, що й отримання рис. 5.9). Як видно, прийнятний рівень розсіювання досягається вже при $\mathfrak{R} > 4$.

Крок 4

На приймальному боці повинні бути відомими первинний ключ $\mathbf{K}_o^*$, алгоритм обчислення ключів **K*** і масив колірності, до якого проводилося вбудовування (**S***). З останнього одержуються значення **X***, **Y***, **N***. Модуль, призначений для видобування прихованого повідомлення, наведено нижче (М.22).

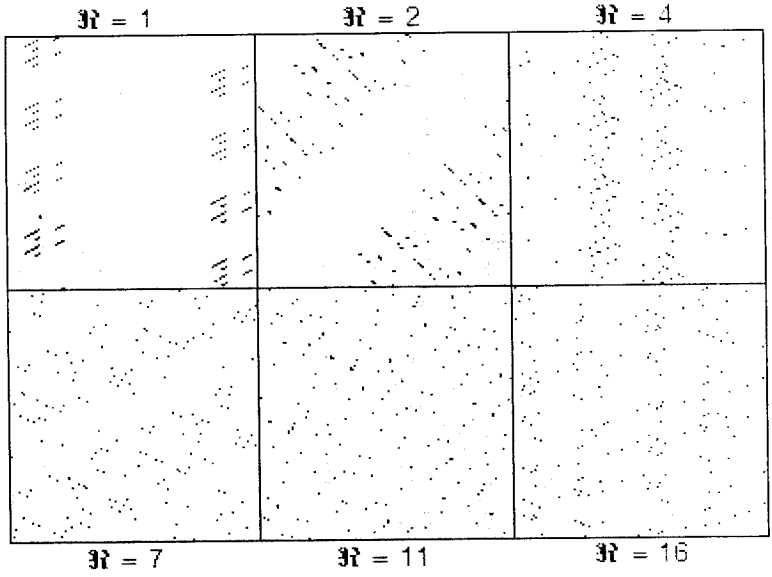


Рис. 5.11. Приклад розсіювання бітів повідомлення по масиву контейнера при зміні параметра $\mathfrak{R}$

Зазначимо, що у даному і подальших методах перед вбудовуванням повідомлення до контейнера додавання до нього стартової і кінцевої міток не проводиться. У такому випадку, при видобуванні текстового рядка **M*** останній, крім корисного змісту, міститиме ще й довільний набір символів, що входять до таблиці кодування *ASCII*. Відсутність етапів виділення корисного повідомлення з усієї множини символів у наступних модулях спричинена намаганням унаочнити безпосередньо процес видобування даних і, водночас, заохоченням читачів до самостійної реалізації того чи іншого способу обмеження основного змісту

стеганограми. За бажанням, наведені модулі можна легко адаптувати для можливості аналізу попередньо вбудованих міток (за аналогією з вищерозглянутими методами — див., наприклад, (М.12) і (М.19)).

```
M*:= for i ∈ 1.. N*/8
       x* ← trunc((i/Y*) + 1),  y* ← mod(i, Y*) + 1
       for s ∈ 1..3
         x* ← mod[x* + B2D[{D2B{K*_(2·s−1), 8} ⊕ D2B(y*, 8)}, X*] + 1
         y* ← mod[y* + B2D[{D2B{K*_(2·s), 8} ⊕ D2B(x*, 8)}, Y*] + 1
       P* ← D2B{S*_(x*, y*), 8}
       M*vec_bin_i ← P*_1
     for j ∈ 1.. rows{M*vec_bin}/8
       M*vec_j ← B2D{submatrix{M*vec_bin, 8·j−7, 8·j, 1, 1}}
       M*vec_j ← 32 if M*vec_j < 32 ∧ M*vec_j ≠ 10 ∧ M*vec_j ≠ 13
     vec2str{M*vec}
```

(М.22)

У нашому випадку результатом видобування є рядок:

M* = "© Alex Puzyrenko, 2017 dâ/@½ÄÁ ßy«æäßO n - ÕýØàb_1Ù?ç¸¡..."

Крок 5

Результати обчислення візуального спотворення контейнера зведені до табл. 5.6 (див. стор. 164).

5.3.2.4. Метод блокового прихування

Метод блокованого прихування є ще одним підходом до реалізації методу заміни і полягає він у наступному [3]. Зображення-оригінал розбивається на I_M блоків Δ_i ($1 \le i \le I_M$), що можуть мати довільну конфігурацію, але при цьому не повинні перетинатися. Для кожного з блоків, на основі аналізу НЗБ кодових комбінацій колірних яскравостей пікселів у їх складі, обчислюється так званий біт парності $b(\Delta_i)$:

$$b(\Delta_i) = \bigoplus_{j \in \Delta_i} LSB(C_j). \qquad (5.4)$$

У кожному блоці проводиться приховування одного секретного біта M_i. Якщо біт парності $b(\Delta_i) \neq M_i$, то відбувається інвертування одного з НЗБ блока Δ_i, у результаті чого $b(\Delta_i) = M_i$. Обрання блока може здійснюватися псевдовипадково з використанням стеганоключа.

Хоча цей метод має таку ж стійкість до спотворень, як і всі попередні, він має низку переваг. По-перше, існує можливість модифікувати значення такого пікселя в блоці, зміна якого призведе до мінімальної зміни статистики контейнера. По-друге, вплив наслідків вбудовування секретних даних у контейнер можна зменшити за рахунок збільшення розміру блока.

Наведемо приклад програми у *Mathcad*, що дозволяє здійснити стеганографічний захист текстового повідомлення методом блокового приховування.

Крок 1

Початкові вихідні дані є типовими і відповідають прийнятим при моделюванні попереднього методу.

Крок 2

Поділ масиву контейнера на блоки здійснюватимемо наступним чином: якщо кількість бітів у повідомленні (**Lм**) не перевищує кількість **Y** стовпців у масиві **C**, то один блок відповідатиме окремому стовпцю масиву **C**. Якщо ж **Lм** > **Y**, то один блок дорівнює $1/\chi$ від окремого стовпця масиву, де χ = **ceil**(**Lм** ÷ **Y**). Значення χ повинне бути відомим одержувачеві.

Алгоритм вбудовування за таким принципом реалізований у модулі (М.23). Початок модуля є аналогічним модулю (М.21). Лічильник **σ** дозволяє виокремлювати відповідну співвідношенню χ частину від загальної довжини стовпця масиву. При цьому визначаються індекси рядків, починаючи з якого (**r1**) і по який (**r2**) виділяється фрагмент **Δ** **y**-го стовпця. Для кожного блока **Δ** проводиться обчислення біта парності **b**. Якщо останній не дорівнює поточному значенню біта повідомлення, з блока **Δ** випадковим чином обирається індекс **n** пікселя, інтенсивність кольору (яскравості) якого збільшується або зменшується на одиницю, в залежності від того, парним чи непарним є первинне його значення. За допомогою функції **putregion**(**S**, **Δ**, **r1**, **y**) здійснюється вбудовування модифікованого масиву **Δ** до загального масиву **S**, починаючи з рядка **r1** і стовпця **y** в бік найстарших індексів рядків і стовпців відповідно.

На рис. 5.12 наведено приклад результату вбудовування — пікселі контейнера, інтенсивність кольору яких зазнала змін (94 пікселі), для наочності ступеня розосередженості стеганошляху показані чорним.

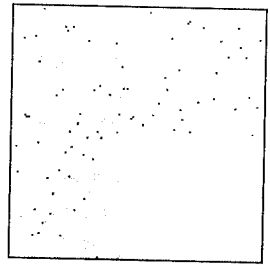


Рис. 5.12. Приклад розсіювання бітів повідомлення по масиву контейнера

$$
\begin{aligned}
S := \;& S \leftarrow C \\
& M_{vec} \leftarrow \mathrm{str2vec}(M),\ M_{vec\_bin} \leftarrow \mathrm{D2B}\left(M_{vec_1}, 8\right) \\
& \text{for } j \in 2 .. \mathrm{rows}\left(M_{vec}\right) \\
& \quad M_{vec\_bin} \leftarrow \mathrm{stack}\left(M_{vec\_bin}, \mathrm{D2B}\left(M_{vec_j}, 8\right)\right) \\
& \text{for } \sigma \in 1 .. \chi \\
& \quad r1 \leftarrow (\sigma - 1) \cdot \mathrm{trunc}\left(\frac{\chi}{\chi}\right) + 1 \\
& \quad r2 \leftarrow \sigma \cdot \mathrm{trunc}\left(\frac{\chi}{\chi}\right) \\
& \quad \text{for } y \in 1 .. Y \\
& \quad\quad \text{break if } y + (\sigma - 1) \cdot Y > L_M \\
& \quad\quad \Delta \leftarrow \mathrm{submatrix}(S, r1, r2, y, y) \\
& \quad\quad b \leftarrow 0 \\
& \quad\quad \text{for } x \in 1 .. \mathrm{rows}(\Delta) \\
& \quad\quad\quad b \leftarrow b \oplus \mathrm{D2B}\left(\Delta_x, 8\right)_1 \\
& \quad\quad \text{if } b \neq M_{vec\_bin_{y+(\sigma-1)\cdot Y}} \\
& \quad\quad\quad n \leftarrow \mathrm{ceil}(\mathrm{rnd}(\mathrm{rows}(\Delta))) \\
& \quad\quad\quad \Delta_n \leftarrow \Delta_n + 1 - 2 \cdot \mathrm{mod}\left(\Delta_n, 2\right) \\
& \quad\quad\quad S \leftarrow \mathrm{putregion}(S, \Delta, r1, y) \\
& S
\end{aligned}
\tag{M.23}
$$

Крок 3

Видобування прихованих даних здійснюється за допомогою програмного модуля (М.24).

```
M* := | for σ ∈ 1 .. χ*                                                    (M.24)
      | r1 ← (σ − 1)·trunc(X* ÷ χ*) + 1
      | r2 ← σ·trunc(X* ÷ χ*)
      | for y ∈ 1 .. Y*
      |   Δ ← submatrix(S*, r1, r2, y, y)
      |   b ← 0
      |   for x ∈ 1 .. rows(Δ)
      |     b ← b ⊕ D2B(Δ_x, 8)_1
      |   M*vec_bin_{y+(σ−1)·Y} ← b
      | for j ∈ 1 .. rows(M*vec_bin) ÷ 8
      |   M*vec_j ← B2D(submatrix(M*vec_bin, 8·j − 7, 8·j, 1, 1))
      |   M*vec_j ← 32  if  M*vec_j < 32 ∧ M*vec_j ≠ 10 ∧ M*vec_j ≠ 13
      | vec2str(M*vec)
```

Видобутий рядок для розглянутого прикладу:

```
M* = "© Alex Puzyrenko, 2017ªSÂ&¾®Ù {·"
```

Крок 4

Результати обчислення візуального спотворення контейнера зведені до табл. 5.6 (див. стор. 164).

5.3.2.5. Методи заміни палітри

Для приховування даних можна також скористатися *палітрою кольорів*, присутніх у форматі зображення [80]. Палітра з N кольорів визначається як список пар індексів (i, Λ_i), який визначає відповідність між індексом i та його вектором колірності Λ_i (так званою *таблицею кольорів*). Кожному пікселю зображення ставиться у відповідність певний індекс у таблиці. Оскільки порядок кольорів у палітрі для відтворення загального зображення не є важливим, конфіденційна інформація може бути прихована шляхом звичайної перестановки кольорів у палітрі.

Для N-кольорової палітри існує $N!$ різних способів перестановки, що для приховання невеликого повідомлення є цілком достатнім. Разом з тим, методи приховування, в основі яких лежить порядок формування палітри, також є нестійкими: будь-яка атака, що має наслідком зміну палітри, руйнує стеганограму.

Сусідні кольори в палітрі найчастіше за все є не схожими один на одного, тому деякі стеганографічні методи перед приховуванням даних проводять упорядкування палітри таким чином, що суміжні кольори стають

подібними. Наприклад, значення кольору може бути впорядковане за відстанню d у RGB-просторі, де $d = \sqrt{R^2 + G^2 + B^2}$ [3]. Оскільки ЗСЛ є більш чутливою до змін яскравості кольору, то вміст палітри доцільно сортувати саме за значеннями яскравості сигналу. Очевидно, що після такого сортування найменш значущі біти у кодах індексів кольорів можна змінювати вже без особливого спотворення зображення.

Деякі стеганометоди [81] передбачають зменшення загальної кількості значень кольорів (до $N/2$) шляхом «розмиття» зображення. При цьому елементи палітри дублюються таким чином, щоб значення кольорів для них розрізнялося несуттєво. У підсумку кожне значення кольору розмитого зображення відповідає двом елементам палітри, які обираються у відповідності до біта приховуваного у зображенні повідомлення.

Для реалізації методу заміни палітри можна запропонувати наступний варіант.

Крок 1

Початкові для вбудовування дані — типові.

Крок 2

Таблицю кольорів контейнера одержимо, використовуючи підмасив інтенсивності червоного **R** — (М.25). Секретність таблиці базується на алгоритмі її формування.

Використовуючи функцію **csort(T, c)**, що дозволяє переставити рядки масиву **T** таким чином, щоб елементи стовпця **c** виявилися відсортованим, упорядкуємо отриману таблицю **T** за інтенсивністю кольорів:

```
T := | i ← 1
     | for x ∈ 1 .. X
     |   for y ∈ 1 .. Y
     |     break if i = 257
     |     Λ ← R[x, y]
     |     on error match(Λ, T^(2))
     |       T[i, 1] ← i
     |       T[i, 2] ← Λ
     |       i ← i + 1
     |   break if i = 257
     | T
```

(М.25)

$$T_{sort} := csort(T, 2).$$

Фрагменти оригінальної та відсортованої колірних таблиць наведено на рис. 5.13.

T =

	1	2
1	1	154
2	2	155
3	3	160
4	4	169
5	5	182
6	6	190
⋮	⋮	⋮
251	251	48
252	252	0
253	253	26
254	254	3
255	255	29
256	256	22

T$_{sort}$ =

	1	2
1	252	0
2	214	1
3	204	2
4	254	3
5	197	4
6	242	5
⋮	⋮	⋮
251	73	250
252	72	251
253	68	252
254	71	253
255	70	254
256	69	255

Рис. 5.13. Оригінальна (**T**) та відсортована (**Tsort**) колірні таблиці

Графічну інтерпретацію колірних таблиць наведено на рис. 5.14.

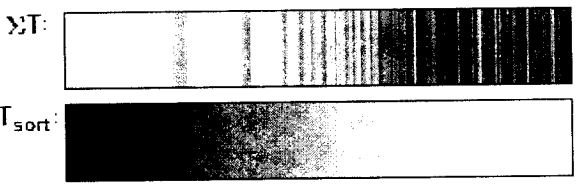


Рис. 5.14. Графічне відображення колірних таблиць **T** і **Tsort**

Крок 3

Модуль (М.26) по вбудовуванню повідомлення до контейнера реалізує наступну послідовність дій:

- формування бітового вектора з символьного рядка аналогічне наведеному у (М.21);
- з масиву контейнера **C** шляхом перебирання індексів рядків (**x**) і стовпців (**y**) змінній **pix** присвоюються значення інтенсивностей кольору відповідних пікселів контейнера;
- функцією **match**(...) проводиться пошук відповідного значення інтенсивності у другому (відсортованому) стовпчику колірної таблиці **Tsort**: у випадку знаходження, змінній **i** присвоюється значення індексу, що відповідає даній інтенсивності у таблиці **Tsort**, а змінній **n** — значення індексу, що відповідає даній інтенсивності у таблиці **T** (1-й стовпчик **Tsort**);

```
S := | (I) – див. (M.21)
     | μ ← 1
     | for x ∈ 1 .. X
     |   for y ∈ 1 .. Y
     |     pix ← C[x, y]
     |     continue on error i ← match(pix, T_sort^(2))1
     |     n ← T_sort[i, 1]
     |     if mod(n, 2) ≠ M_vec_bin[μ]
     |       λL ← −1000 , λH ← 1000
     |       for σ ∈ 1 .. 255
     |         if mod(T_sort[i−σ, 1], 2) = M_vec_bin[μ]  if i − σ ≥ 1
     |           λL ← T_sort[i−σ, 2]
     |           break
     |       for σ ∈ 1 .. 255
     |         if mod(T_sort[i+σ, 1], 2) = M_vec_bin[μ]  if i+σ ≤ 256
     |           λH ← T_sort[i+σ, 2]
     |           break
     |       S[x, y] ← if(pix − λL ≤ λH − pix, λL, λH)
     |     μ ← μ + 1
     |     break if μ > L_M
     |   break if μ > L_M
     | S
```

(M.26)

- якщо НЗБ індексу **n** не дорівнює поточному біту приховуваного повідомлення, відбувається пошук найближчого індексу, для якого така рівність виконується: пошук ведеться вниз (**L**) і вгору (**H**) від індексу **i**[X];
- після знаходження значень λ_L і λ_H пікселю контейнера **S** присвоюється те з них, яке по колірній вісі знаходиться найближче до інтенсивності відповідного пікселя контейнера **C** (**pix**);
- після вбудовування останнього біта повідомлення зовнішній цикл переривається — контейнер заповнений.

[X] Попереднє присвоєння змінним λ_L та λ_H значення −/+1000 гарантує неможливість дублювання попередніх значень λ, якщо просування вниз або вгору від індексу **i** не призвело до виконання поставленої умови (останнє, в принципі, є можливим при знаходженні індексу **i** занадто близько до нижньої або верхньої межі відсортованої колірної таблиці).

Крок 4

При видобуванні повідомлення необхідно сформувати таблиці кольорів **T*** і **T*sort** на основі масиву **R***. Модуль, що реалізує дану операцію, є ідентичним (М.25).

Крок 5

Модуль видобування (М.27) для інтенсивності кожного пікселя масиву **S*** проводить пошук відповідної інтенсивності у колірній таблиці. При знаходженні, μ-му елементу бітового повідомлення **M*** присвоюється значення НЗБ індексу, що відповідає даній інтенсивності у невідсортованій таблиці. Одержаний бітовий вектор в кінці модуля переводиться на рядок символів.

```
M* := | μ ← 1                                                          (М.27)
      | for x ∈ 1 .. X*
      |   for y ∈ 1 .. Y*
      |     | pix ← S*x,y
      |     | continue on error | i ← match(pix, T*sort^(2))1
      |     |                   | M*μ ← mod(T*sort_i,1, 2)
      |     |                   | μ ← μ + 1
      | for j ∈ 1 .. rows(M*) ÷ 8
      |   | M*vec_j ← B2D(submatrix(M*, 8·j − 7, 8·j, 1, 1))
      |   | M*vec_j ← 32 if M*vec_j < 32 ∧ M*vec_j ≠ 10 ∧ M*vec_j ≠ 13
      | vec2str(M*vec)
```

Рядок, що при цьому було видобуто:

M* = "© Alex Puzyrenko, 2017WóÒK{ëÿÿnã ¬TμÉâ Û AúÿÇãÕ…"

Крок 6

Результати обчислення візуального спотворення контейнера зведені до табл. 5.6 (див. стор. 164).

5.3.2.6. Метод квантування зображення

До методів приховання в просторовій області можна також віднести і *метод квантування зображення* [3, 82], заснований на між-

піксельній залежності, що може бути описаною через деяку функцію Θ. У найпростішому випадку можна обчислити різницю ε_i між суміжними пікселями c_i та c_{i+1} (або c_{i-1} та c_i), задавши її в якості параметра функції Θ:

$$\Delta_i = \Theta(c_i - c_{i+1}),$$

де Δ_i — дискретна апроксимація різниці сигналів $c_i - c_{i+1}$.

Оскільки Δ_i є цілим числом, а реальна різниця $c_i - c_{i+1}$ — дійсним, то виникають помилки квантування $\delta_i = \Delta_i - \varepsilon_i$. Для сильно корельованих сигналів ця помилка є близькою до нуля: $\delta_i \approx 0$.

У даному методі приховування інформації здійснюється шляхом корегування різницевого сигналу Δ_i. Стеганоключ являє собою таблицю, в якій кожному можливому значенню Δ_i ставиться у відповідність визначений біт, наприклад:

Δ_i	–4	–3	–2	–1	0	1	2	3	4
k_i	1	0	1	1	0	0	1	0	1

Для приховування i-го біта повідомлення обчислюється різниця Δ_i. Якщо при цьому k_i не відповідає секретному бітові, який необхідно приховати, то значення Δ_i замінюється найближчим Δ_j, для якого така умова виконується. При цьому значення інтенсивностей пікселів, між якими обраховувалася різниця Δ_i, відповідним чином корегуються. Видобування секретного повідомлення проводиться відповідно до значення k_i^*, що відповідає різниці Δ_i^*.

Розглянемо приклад програми, що реалізує стеганографічний метод квантування зображення.

Крок 1

Первинні дані є типовими.

Крок 2

Стеганоключ обчислимо за модулями (М.28) і (М.29). При цьому модуль (М.28) повертає всі можливі різниці сигналів (від –255 до +255), а модуль (М.29) – – значення бітів, що цим різницям відповідають.

```
Δ := | for i ∈ 1 .. 511
     |   Δ_i ← i – 256
     | Δ
```
(М.28)

```
k := | for y ∈ 1..Y                                   (M.29)
     |     c ← R^⟨y⟩
     |     k_y ← 1  on error k_y ← ¬k_{y−1}
     |     for x ∈ 1..X
     |         k_y ← k_y ⊕ mod(c_x, 2)
     | while rows(k) < rows(Δ)
     |         k ← stack(k, k)
     | k
```

У даному випадку стеганоключ **k** формується на основі масиву червоної колірної складової. При цьому для кожного стовпця масиву **R** обчислюється сума за модулем 2 усіх елементів стовпця з додаванням результату до інверсного значення попереднього елемента (або до 1, якщо формується тільки перший розряд ключа).

Наприкінці модуля одержаний вектор **k**, елементи якого носять псевдовипадковий характер, розширюється на довжину вектора **Δ**. Приклад сформованого стеганоключа наведено на рис. 5.15.

Δ =

1	-255
2	-254
3	-253
4	-252
⋮	⋮
254	-2
255	-1
256	0
257	1
258	2
⋮	⋮
508	252
509	253
510	254
511	255

k =

1	0
2	1
3	1
4	1
⋮	⋮
254	1
255	1
256	1
257	0
258	1
⋮	⋮
509	1
510	1
511	1
512	1

rows(**Δ**) = 511

rows(**k**) = 512

mean(**Δ**) = 0

mean(**k**) = 0.531

Рис. 5.15. Фрагменти різницевого сигналу **Δ** і прикладу стеганоключа **k**, потужності (**rows**) і середні значення (**mean**) відповідних векторів

Крок 3

Використовуючи модуль (М.15), проведемо розгортання масиву контейнера **C** (масив синьої колірної складової **B**) у вектор **Cv**.

Задамося стартовим індексом елемента отриманого вектора, починаючи з якого проводитиметься вбудовування бітів повідомлення (наприклад, $\mu_s := 23$). Для підрахунку величини кроку (псевдовипадкового інтервалу) між вбудовуваннями скористаємося модулем (М.14); нехай при цьому **K** := 11,89.

Крок 4

Алгоритм вбудовування реалізує програмний модуль (М.30). Формування вектора двійкових даних з рядка символів є аналогічним наведеному у (М.21), але при цьому запис **S** ← **C** слід замінити на **Sv** ← **Cv**.

```
Sv := | (I) – див. (М.21)                                                    (М.30)
      | z ← μ_s
      | for μ ∈ 1.. L_M
      |   z ← z + step(D2B(z, ceil(log(z, 2))))
      |   Δ' ← Cv_z – Cv_(z–1)
      |   continue  on error  i ← match(Δ', Δ)_1
      |   if  k_i ≠ M_vec_bin_μ
      |     | l_L ← –1000 , l_H ← 1000
      |     | for  σ ∈ 1.. i – 1
      |     |   if  k_(i–σ) = M_vec_bin_μ
      |     |     | l_L ← i – σ
      |     |     | break
      |     | for  σ ∈ 1.. rows(Δ) – i
      |     |   if  k_(i+σ) = M_vec_bin_μ
      |     |     | l_H ← i + σ
      |     |     | break
      |     | if  i – l_L < l_H – i
      |     |     | Sv_z ← Sv_(z–1) + Δ_(l_L)   if  0 ≤ Sv_(z–1) + Δ_(l_L) ≤ 255
      |     |     | Sv_(z–1) ← Sv_z – Δ_(l_L)   otherwise
      |     | otherwise
      |     |     | Sv_z ← Sv_(z–1) + Δ_(l_H)   if  0 ≤ Sv_(z–1) + Δ_(l_H) ≤ 255
      |     |     | Sv_(z–1) ← Sv_z – Δ_(l_H)   otherwise
      | Sv
```

Для кожного μ-го біта повідомлення проводиться обчислення індексу **z** елемента вектора контейнера **Cv**. Обчислюється різниця **Δ'** між суміжними пікселями $\mathbf{Cv_z}$ і $\mathbf{Cv_{z-1}}$ і за допомогою функції **match**(...) здійснюється пошук індексу **i** відповідного **Δ'** значення різниці у векторі **Δ**.

Якщо значення $\mathbf{k_i}$ не дорівнює поточному приховуваному біту, відбувається пошук найближчого індексу, при якому $\mathbf{k_i}$ дорівнює цьому бітові. Пошук проводиться вниз (**L**) і вгору (**H**) від індексу **i**. Попереднє присвоєння змінним $\mathbf{I_L}$ та $\mathbf{I_H}$ значень ±1000 забезпечує неможливість дублювання попередніх значень **i**, якщо рух вниз або вгору від **i** не призвів до виконання поставленої умови (останнє є можливим при знаходженні індексу **i** занадто близько до нижньої або верхньої межі вектора **k**).

Після знаходження значень $\mathbf{I_L}$ та $\mathbf{I_H}$ обирається те з них, яке є найближчим до **i**. При цьому інтенсивність пікселя контейнера $\mathbf{Sv_z}$ дорівнює збільшеній на величину $\mathbf{\Delta_{I_{L(H)}}}$ інтенсивності суміжного пікселя $\mathbf{Sv_{z-1}}$. Якщо дане збільшення призводить до виходу значення інтенсивності кольору за межі [0; 255], то, навпаки, інтенсивності суміжного пікселя $\mathbf{Sv_{z-1}}$ присвоюється значення інтенсивності пікселя $\mathbf{Sv_z}$, зменшеній на величину $\mathbf{\Delta_{I_{L(H)}}}$. Після вбудовування останнього біта повідомлення ($\mathbf{L_M}$) зовнішній цикл переривається.

Крок 5

Проводимо зворотне згортання вектора у матрицю, що має розмірність первинного масиву **C** — див. (М.7). Одержуємо масив **S**.

Отриманий у нашому прикладі стеганошлях (без урахування тих пікселів, яскравість яких залишилася незмінною, бо вже відповідала приховуваним бітам повідомлення) наведений на рис. 5.16.

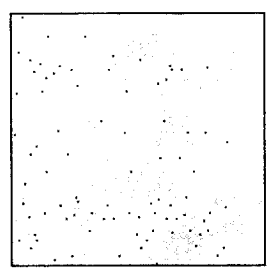


Рис. 5.16. Біти стеганошляху в межах контейнера

Крок 6

При видобуванні повідомлення попередньо формується стеганоключ — пара векторів Δ^* і k^*. Програмні модулі при цьому є ідентичними (М.28) і (М.29). Масив контейнера розгортається у вектор Sv^* (подібно до (М.15)).

Крок 7

Програмний модуль видобування (М.31) обчислює різницю інтенсивностей суміжних пікселів Sv^*_z та Sv^*_{z-1} і здійснює пошук індексу i відповідного Δ' значення різниці у векторі Δ^*.

$$
\begin{aligned}
M^* := \;& z \leftarrow \mu^*_s \\
& \text{for } \mu \in 1 .. \text{rows}(Sv^*) \\
& \quad z \leftarrow z + \text{step}(D2B(z, \text{ceil}(\log(z,2)))) \\
& \quad \Delta' \leftarrow Sv^*_z - Sv^*_{z-1} \text{ if } z \le \text{rows}(Sv^*) \\
& \quad \text{break if } z > \text{rows}(Sv^*) \\
& \quad \text{continue on error } i \leftarrow \text{match}(\Delta', \Delta^*)_1 \\
& \quad M^*_\mu \leftarrow k^*_i \\
& \text{for } j \in 1 .. \text{rows}(M^*) \div 8 \\
& \quad M^*_{vec_j} \leftarrow B2D(\text{submatrix}(M^*, 8\cdot j - 7, 8\cdot j, 1, 1)) \\
& \quad M^*_{vec_j} \leftarrow 32 \text{ if } M^*_{vec_j} < 32 \wedge M^*_{vec_j} \neq 10 \wedge M^*_{vec_j} \neq 13 \\
& \text{vec2str}(M^*_{vec})
\end{aligned}
\qquad \text{(М.31)}
$$

Значення біта k^*_i, що, згідно індексу i, відповідає даній різниці, присвоюється поточному (μ-му) елементові вектора M^*. Наприкінці модуля отриманий вектор двійкових даних перетворюється на символьний рядок. У нашому випадку було отримано наступне:

$$M^* = \text{"© Alex Puzyrenko, 2017øá"}$$

Крок 8

Одержані при обчисленні візуального спотворення результати зведено до табл. 5.6 (див. стор. 164).

5.3.2.7. Метод Куттера-Джордана-Боссена

Мартін Куттер *(M. Kutter)*, Фредерік Джордан *(F. Jordan)* та Френк Боссен *(F. Bossen)* [83] запропонували алгоритм вбудовування до каналу синього кольору *RGB*-зображення, оскільки саме до нього є найменш чутливою ЗСЛ.

Розглянемо алгоритм передавання окремого біта секретної інформації у запропонованому методі. Нехай m_i — біт, що підлягає вбудовуванню; $\mathbf{C} = \{\mathbf{R}, \mathbf{G}, \mathbf{B}\}$ — зображення-контейнер; $p = (x, y)$ — псевдовипадково обраний піксель контейнера (канал синього кольору), до якого виконуватиметься вбудовування біта m_i шляхом модифікації яскравості пікселя $\lambda_{x,y} = 0.29890 \cdot R_{x,y} + 0.58662 \cdot G_{x,y} + 0.11448 \cdot B_{x,y}$:

$$B'_{x,y} = \begin{cases} B_{x,y} - \upsilon \cdot \lambda_{x,y}, & \text{при } m_i = 0; \\ B_{x,y} + \upsilon \cdot \lambda_{x,y}, & \text{при } m_i = 1 \end{cases} = B_{x,y} + (2 \cdot m_i - 1) \cdot \upsilon \cdot \lambda_{x,y}, \qquad (5.5)$$

де υ — константа, що визначає енергію вбудовуваного сигналу. Її величина залежить від призначення стеганосистеми. Чим більшою є υ, тим вище стійкість вбудованої інформації до спотворень, але й тим сильнішою буде відмінність заповненого контейнера від оригіналу.

Видобування біта одержувачем здійснюється без наявності у нього первинного зображення, тобто «наосліп». Для цього виконується передбачення значення первинного, немодифікованого пікселя на основі значень сусідніх із ним пікселів. Для одержання такої оцінки пікселя запропоновано використовувати значення тих декількох пікселів, що розміщені у тому ж стовпці й у тому ж рядку масиву графічного контейнера. Зокрема, автори [83] використовували конфігурацію з пікселів у формі «плюса» розміром 7×7. У цьому випадку оцінка $\hat{B}^*_{x,y}$ отримується шляхом обчислення наступного виразу:

$$\hat{B}^*_{x,y} = \frac{1}{4 \cdot \sigma} \cdot \left[\sum_{i=-\sigma}^{+\sigma} B^*_{x+i,y} + \sum_{j=-\sigma}^{+\sigma} B^*_{x,y+j} - 2 \cdot B^*_{x,y} \right], \qquad (5.6)$$

де σ — це кількість пікселів згори (знизу, ліворуч, праворуч) від оцінюваного пікселя (у випадку «плюса» 7×7 $\sigma = 3$).

При видобуванні прихованого біта обчислюється різниця δ між поточним ($B^*_{x,y}$) і прогнозованим ($\hat{B}^*_{x,y}$) значеннями інтенсивності пікселя $p = (x, y)$:

$$\delta = B^*_{x,y} - \hat{B}^*_{x,y}. \qquad (5.7)$$

Знак, що матиме у підсумку δ, означатиме біт, який було вбудовано: якщо $\delta < 0$, то m_i = «0»; якщо $\delta > 0$, то m_i = «1».

Функції вбудовування і видобування у даному методі є асиметричними — функція видобування не є зворотною по відношенню до функції вбудовування. Хоча, як зазначають автори, правильне роз-

пізнання біта повідомлення у випадку застосування описаних вище процедур, є високоймовірним, проте, все ж таки, не стовідсотково вірним. Для зменшення ймовірності помилок при видобуванні було запропоновано багаторазово дублювати вбудовування кожного біта. Оскільки при цьому кожен біт повторюється τ разів, то одержувач має змогу дістати τ його оцінок. Остаточне значення секретного біта отримується після усереднення різниці між реальним (поточним) та оціненим (прогнозованим) значеннями інтенсивності пікселя в досліджуваному контейнері:

$$\delta = \tau^{-1} \cdot \sum_{i=1}^{\tau} \left[B^{*}_{x,y} - \hat{B}^{*}_{x,y} \right]. \qquad (5.8)$$

Як і у випадку (5.7), знак усередненої різниці визначатиме значення вбудованого біта. У роботі [83] показано, що пропонований алгоритм є стійким до багатьох з відомих атак: НЧ фільтрації зображення, його стиснення згідно алгоритму *JPEG*, обрізанню країв тощо.

У рамках цієї книги пропонується наступна реалізація даного методу.

Крок 1

Первинні дані — типові.

Крок 2

Матриця яскравості пікселів графічного контейнера одержується використанням функції **READBMP**("ім'я_файлу"), що повертає масив, який представляє зображення *BMP*-формату у яскравісному аспекті (градаціях сірого): λ := **READBMP**("C.bmp").

Крок 3

Алгоритм вбудовування реалізується програмним модулем (М.32). При цьому обчислення псевдовипадкових координат пікселя, до якого відбуватиметься вбудовування біта повідомлення виконується за алгоритмом, описаним у підпункті 5.3.2.3 для методу псевдовипадкової перестановки (див. (М.21)).

Попередньо задаються параметр υ, що визначає енергію вбудованого сигналу, та кількість повторювань вбудовування одного й того ж самого біта — τ. Встановлено, що результат вбудовування буде візуально непомітним при значеннях $\upsilon < 0{,}05$. Проте, у такому випадку для зменшення помилок при видобуванні доведеться значно підвищувати кількість приховувань τ поточного біта ($\tau > 35$), що теж негативно відбивається на статистиці зображення. Оптимальними, на наш погляд, є значення $\upsilon \approx 0.15$ і $\tau \le 30$, але, знову ж таки, все залежатиме від характеристик зображення, що було обране в якості контейнера.

$$
\begin{aligned}
B' := \Big| & \textcircled{I} \ - \text{див. (М.21); замість } S \leftarrow C \text{ використати } B' \leftarrow B \qquad \text{(М.32)} \\
& \text{for } i \in 1..\tau \cdot L_M \\
& \quad x \leftarrow \text{trunc}\left(\frac{i}{Y}\right) + 1,\ y \leftarrow \text{mod}(i, Y) + 1 \\
& \quad \text{for } s \in 1..\mathfrak{R} \\
& \quad\quad x \leftarrow \text{mod}\left[x + B2D\left[\overrightarrow{(D2B(K_{2\cdot s-1}, 8) \oplus D2B(y, 8))}, X\right] + 1\right] \\
& \quad\quad y \leftarrow \text{mod}\left[y + B2D\left[\overrightarrow{(D2B(K_{2\cdot s}, 8) \oplus D2B(x, 8))}, Y\right] + 1\right] \\
& \quad j \leftarrow \text{ceil}\left(\frac{i}{\tau}\right) \\
& \quad B'_{x,y} \leftarrow B_{x,y} + \left(2 \cdot M_{vec\_bin_j} - 1\right) \cdot \upsilon \cdot \lambda_{x,y} \\
& \text{scale}(B', 0, 255)
\end{aligned}
$$

Обчислення індексу елемента двійкового вектора повідомлення за формулою **ceil**(**i**/τ) дозволяє один і той самий біт повідомлення (**j**-й) приховати рівно τ разів, після чого настає черга наступного біта і т. д., аж допоки не відбудеться $\tau \cdot L_M$ вбудовувань.

Після проведення модифікації інтенсивності пікселя, визначеного координатами (**x**, **y**), проводиться корегування значення результуючої інтенсивності за допомогою функції **scale**(**A**, **Min**, **Max**), що повертає масив **A**, елементи якого є віднормованими у діапазоні [**Min**, **Max**]. Інакше, при початковому значенні інтенсивності кольору пікселя, наприклад, 255, внесення біта повідомлення «1» призведе не до зростання значення інтенсивності, а навпаки, — до її зменшення в бік темних відтінків (при формуванні зображення усі числа, що перевищують 255, автоматично замінюються на залишок від свого ділення на число 256). За тим самим принципом, при занадто низьких початкових значеннях інтенсивності пікселя, зокрема 0, внесення до нього біта «0» може призвести до одержання від'ємного значення, яке, у свою чергу, розумітиметься програмою як значення інтенсивності, близьке до 255.

Крок 4

Перед видобуванням повідомлення мають бути відомими: параметри контейнера, первинний ключ K_o^*, кількість раундів обчислення координат (**x**, **y**) — $\mathfrak{R}^*$, кількість дублюючих вбудовувань окремого біта τ^*, а також розмірність «плюса» σ — кількість пікселів згори (знизу, ліворуч, праворуч) від оцінюваного пікселя.

Модуль видобування прихованого повідомлення (М.33) містить у собі блок обчислення псевдовипадкових координат (**x**, **y**), ідентичний

відповідному у модулі вбудовування (М.32). Далі йдуть блоки виконання умов, що в сукупності дозволяють врахувати проблемні випадки, коли оцінюваний піксель знаходиться занадто близько до краю(-їв) зображення і побудувати повноцінний «плюс» заданої розмірності з оточуючих пікселів не виявляється можливим (див. рис. 5.17).

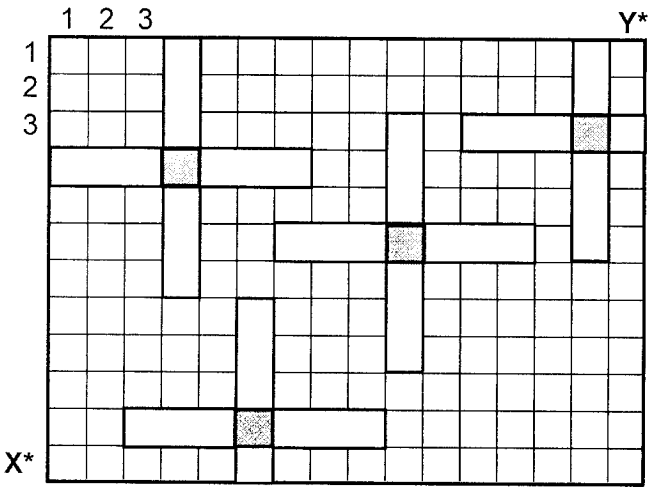


Рис. 5.17. Приклади оцінюваних пікселів і оцінювальних конфігурацій («плюсів» 7×7: $\sigma = 3$)

У наведеному модулі попередньо проводиться генерування псевдовипадкових індексів (**x**, **y**), які визначатимуть елемент масиву **B***, навколо якого робитиметься оцінка прилеглих пікселів. За результатами генерування даних індексів обчислюється кількість пікселів згори і знизу, ліворуч і праворуч від оцінюваного. У подальшому проводиться обчислення оцінки первинного значення оцінюваного пікселя (формула (5.6)). Одержаний результат заноситься до **t**-го елемента **B*^**. Оцінюване значення пікселя $B^*_{x,y}$ зберігається у буферному масиві B^*_t (велика бета).

Поки $t \le \tau^*$, продовжується накопичення оцінок пікселів, до яких було вбудовано один і той самий біт повідомлення. Коли ж $t > \tau^*$, підсумки попереднього оцінювання узагальнюються: обчислюється усереднена різниця δ між первинними та оціненими значеннями інтенсивностей пікселів, що виступили контейнерами для одного біта вбудовуваних даних (формула (5.8)). У залежності від знаку одержаної

різниці j-му елементові вектора двійкових даних (індекс елемента визначається за результатом обчислення функції **ceil**(i/τ*)) присвоюється значення 0 або 1. Змінна **t** скидається в 1 і починається збір оцінок значень інтенсивностей пікселів, до яких було вбудовано наступний біт повідомлення. Процес повторюється до тих пір, поки не буде проаналізовано всі елементи графічного масиву.

$$
\begin{aligned}
&M^* := \left| \begin{array}{l}
t \leftarrow 1 \\
\text{for } i \in 1..\, X^* \cdot Y^* \\
\quad x \leftarrow \text{trunc}(i \div Y^*) + 1,\ y \leftarrow \text{mod}(i, Y^*) + 1 \\
\quad \text{for } s \in 1..\, \mathfrak{R}^* \\
\quad\quad x \leftarrow \text{mod}\left[x + B2D\left[\overrightarrow{\left(D2B\left(K^*_{2\cdot s-1}, 8\right) \oplus D2B(y, 8)\right)}, X^*\right]\right] + 1 \\
\quad\quad y \leftarrow \text{mod}\left[y + B2D\left[\overrightarrow{\left(D2B\left(K^*_{2\cdot s}, 8\right) \oplus D2B(x, 8)\right)}, Y^*\right]\right] + 1 \\
\quad \sigma 1 \leftarrow -\sigma,\ \sigma 2 \leftarrow \sigma \ \text{ if } \ \sigma < x \le X^* - \sigma \\
\quad \text{otherwise} \\
\quad\quad \sigma 1 \leftarrow 1 - x,\ \sigma 2 \leftarrow \sigma \ \text{ if } \ x \le \sigma \\
\quad\quad \sigma 1 \leftarrow -\sigma,\ \sigma 2 \leftarrow X^* - x \ \text{ otherwise} \\
\quad \sigma 3 \leftarrow -\sigma,\ \sigma 4 \leftarrow \sigma \ \text{ if } \ \sigma < y \le Y^* - \sigma \\
\quad \text{otherwise} \\
\quad\quad \sigma 3 \leftarrow 1 - y,\ \sigma 4 \leftarrow \sigma \ \text{ if } \ y \le \sigma \\
\quad\quad \sigma 3 \leftarrow -\sigma,\ \sigma 4 \leftarrow Y^* - y \ \text{ otherwise} \\
\quad B^{*\wedge}{}_t \leftarrow \dfrac{\sum_{a=\sigma 1}^{\sigma 2} B^*_{x+a,\,y} + \sum_{a=\sigma 3}^{\sigma 4} B^*_{x,\,y+a} - 2 \cdot B^*_{x,\,y}}{\sigma 2 - \sigma 1 + \sigma 4 - \sigma 3} \\
\quad B^*_t \leftarrow B^*_{x,\,y} \\
\quad t \leftarrow t + 1 \\
\quad \text{if } \ t > \tau^* \\
\quad\quad \delta \leftarrow \dfrac{1}{\tau^*} \cdot \sum_{T=1}^{\tau^*} \left(B^*_T - B^{*\wedge}{}_T\right) \\
\quad\quad j \leftarrow \text{ceil}(i \div \tau^*) \\
\quad\quad M^*_{vec\_bin_j} \leftarrow \text{if}(\delta \le 0, 0, 1) \\
\quad\quad t \leftarrow 1 \\
\text{for } j \in 1..\, \text{rows}\left(M^*_{vec\_bin}\right) \div 8 \\
\quad M^*_{vec_j} \leftarrow B2D\left(\text{submatrix}\left(M^*_{vec\_bin}, 8 \cdot j - 7, 8 \cdot j, 1, 1\right)\right) \\
\quad M^*_{vec_j} \leftarrow 32 \ \text{ if } \ M^*_{vec_j} < 32 \wedge M^*_{vec_j} \ne 10 \wedge M^*_{vec_j} \ne 13 \\
\text{vec2str}\left(M^*_{vec}\right)
\end{array} \right. \qquad \text{(M.33)}
\end{aligned}
$$

Видобутий рядок для випадку, коли $\upsilon = 0.15$, $\tau = 25$, а оцінювальним елементом є «плюс» пікселів з $\sigma = 3$:

$$M^* = \text{"© Alex Puzyrenko, 2017{{Ãi5?°/¾v§..."}}$$

Крок 4

Порівняння пустого і заповненого контейнерів наведене на рис. 5.18.

Результати обчислення візуального спотворення контейнера зведені до табл. 5.6 (див. стор. 164).

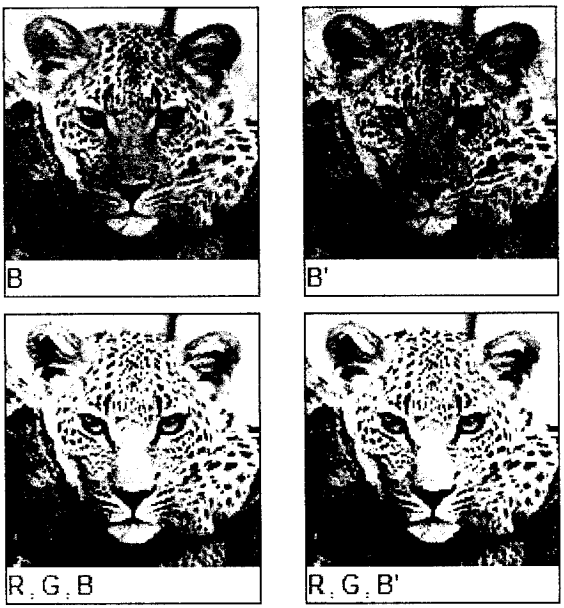


Рис. 5.18. Пустий (ліворуч) і заповнений (праворуч) контейнери (канал синього кольору — верхній рядок; *RGB*-зображення — нижній)

Крок 5

Скористаємося показником *BER* (*Bit Error Rate*) — відношенням кількості бітів, прийнятих з помилкою, до загальної кількості переданих бітів (у нашому випадку $L_M = 8 \cdot 22 = 176$ бітів), — і дослідимо залежність якості видобутої стеганограми від розміру σ «плюса», «ікса», «зірки» і «квадрата» з навколишніх для кожного контейнера стеганошляху пікселів (при $\upsilon = 0.15$, $\tau = 25$). Результат зведено до табл. 5.1.

Таблиця 5.1

Залежність *BER* від розміру конфігурації навколишніх пікселів

σ, *пікс.*		1	2	3	5	7	11	15	25	50
BER, %	«+»	0.000	0.000	0.000	0.000	0.000	0.568	0.568	1.705	3.409
	«×»	0.000	0.000	0.000	2.841	3.977	6.250	6.818	21.023	33.523
	«✳»	0.000	0.000	0.000	0.000	0.000	0.568	1.136	1.705	6.250
	«□»	0.000	0.000	0.568	2.273	3.977	11.932	17.045	30.114	

Крок 6

Для випадку σ = 3, υ = 0.15, τ = 25, дослідимо вплив стиснення *JPEG* на можливість видобування прихованих даних — табл. 5.2.

Таблиця 5.2

Залежність *BER* від рівня збереженої якості при *JPEG* стисненні[XI]

Якість, %		100	98	96	93	90	86	82	75	60	40
Розмір файлу, *КіБ*		34.2	25.1	21.3	17.6	15.8	12.8	11.2	9.8	7.6	6.1
BER, %	«+»	0.000	0.000	0.000	0.000	0.000	5.682	19.318	28.977	36.364	41.477
	«×»	0.000	0.000	0.000	0.000	1.705	6.818	19.886	30.682	35.795	39.205
	«✳»	0.000	0.000	0.000	0.000	0.000	5.114	18.750	30.682	36.932	41.477
	«□»	0.000	0.000	0.000	0.000	1.136	8.523	24.432	35.795	39.773	

Як бачимо, при невеликих значеннях υ і τ стиснення з більш як 15...20% падінням якості (що є рівноцінним зменшенню розміру файлу приблизно у 4 рази) кількість помилково розпізнаних бітів суттєво зростає. Тим не менш, з проведених дослідів було встановлено, що в деяких випадках суттєве збільшення кількості повторень τ може зробити стеганограму стійкою (*BER* = 0%) навіть при *JPEG* стисненні з 0% збереження якості (стиснення більш як у 10 разів) — табл. 5.3.

[XI] За основу беруться дані графічного редактора *ACD FotoCanvas* розробника *«ACD Systems International Inc.»*:

Таблиця 5.3

Залежність *BER* від кількості повторень вбудовування окремого біта при 0% збереженої якості стиснутого зображення[XII]

т. разів		10	20	33	60	100	200	333	666	1000	2000
Довжина стегано-грами, *симв.*		200	100	60	33	200	10	6	3	2	1
BER, %	«+»	40.56	41.25	37.08	27.65	25.0	16.25	18.75	0.00	0.00	0.00
	«×»	40.25	38.88	34.58	25.38	23.75	15.00	10.42	0.00	0.00	0.00
	«✳»	40.75	40.00	32.71	24.24	23.75	15.00	12.50	0.00	0.00	0.00
	«□»	[illegible]	42.25	36.46	29.17	28.75	27.5	33.33	25.00	31.25	37.50

5.3.2.8. Метод Дармстедтера-Делейгла-Квісквотера-Мака

Нетривіальне блокове вбудовування до просторової області графічного контейнера свого часу запропонували Вінсент Дармстедтер *(V. Darmstaedter)*, Жан-Франсуа Делейгл *(J.-F. Delaigle)*, Жан-Жак Квісквотер *(J.-J. Quisquater)* та Бенуа Мішель Мак *(B. M. Macq)* [100]. Розроблений ними метод дозволяє досягти компромісу між стійкістю стеганосистеми до спотворювань, якістю вбудовування і, зрештою, обчислювальною складністю алгоритму. Сам метод базується на елементарному перцепційному (відчуттєвому) сприйнятті і дозволяє пристосовувати вбудовування відносно поточного вмісту блоків.

Перед вбудовуванням, конфіденційна інформація перетворюється на вектор двійкових даних, для розміщення кожного біта з якого використовується окремий блок. У розгляданому авторами варіанті розмірність блоків становила 8×8 пікселів. Головною причиною такого вибору була сумірність з блоками, що використовуються під час *JPEG*-стиснення, і, як наслідок, дія останньої однаковим чином позначатиметься на кожному з вбудованих бітів. Крім того, приховувану інформацію вбудовують з деякою надмірністю, що підвищує загальну стійкість стеганосистеми.

У загальному випадку процес вбудовування бітів повідомлення здійснюється у чотири етапи:

1) поділ масиву зображення-контейнера на блоки 8×8 пікселів;
2) класифікація пікселів окремого блока на *зони* з приблизно однорідними значеннями яскравості;

[XII] Розмір стиснутого файлу — 4.5 КіБ.

3) поділ кожної зони на *категорії* у відповідності до індивідуальної (псевдовипадкової) маски;

4) вбудовування біта в залежності від співвідношення між середніми значеннями категорій кожної зони шляхом модифікації значень яскравості кожної категорії в кожній зоні.

Розглянемо останні три етапи докладніше.

Класифікація пікселів на зони

Мета етапу полягає у тому, щоб розбити пікселі всередині блока на групи, що мали б приблизно однакову яскравість. Така класифікація бере до уваги особливості блока, що викликають інтерес з точки зору непомітності та стійкості. При класифікації автори виділили три типи контрасту:

- *виразний контраст* (рис. 5.19, *а*), коли можна розрізнити дві зони, розділені різким стрибком яскравості;
- *поступовий контраст* (рис. 5.19, *б*), коли дві однорідні зони розділені поступовою зміною яскравості;
- *шумовий (нечіткий) контраст* (рис. 5.19, *в*) з яскравістю, розподіленою на зразок випадкового шуму (у граничному випадку шумовий контраст вироджується в однотонове зображення — контраст як такий відсутній, усі пікселі блока мають однакову яскравість).

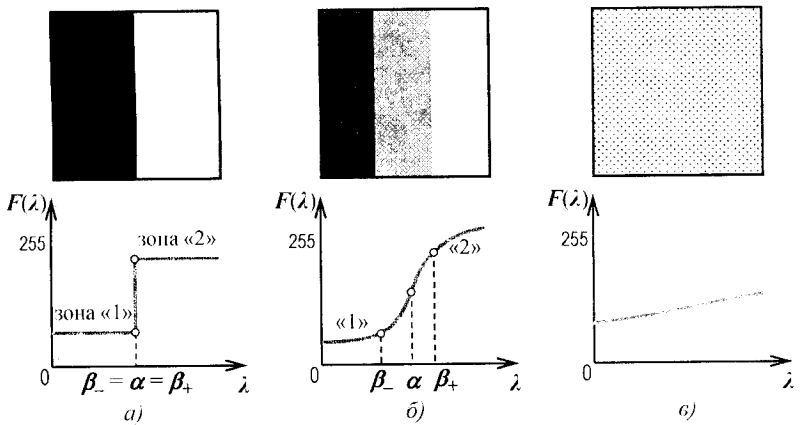


Рис. 5.19. Класифікація на зони: виразний контраст (*а*), поступовий контраст (*б*), шумовий контраст (*в*)

Відсортовані за зростанням, значення яскравостей пікселів блока можна представити зростаючою функцією $F(\lambda)$, де $F(1)$ і $F(N^2)$ являють собою відповідно найменше і найбільше значення яскравості серед усіх, наявних у даному блоці (де N — розмірність квадратного блока).

Тип контрасту блока визначає крутизна функції $F(\lambda)$, яку позначимо через $S(\lambda)$. Нехай S_{max} — це максимальна крутизна функції F, при $\lambda = \alpha$. Якщо S_{max} нижче за деякий заданий поріг T_1, вважається, що блок має шумовий контраст. Якщо ж S_{max} вище за поріг T_1, блок має або поступовий, або виразний контраст. У цьому випадку додатково визначають параметри β_+ і β_- — індекси в найближчому околі точки α (відповідно вище і нижче за неї), що задовольняють нерівностям

$$S(\alpha) - S(\beta_+) > T_2 \quad \text{і} \quad S(\alpha) - S(\beta_-) > T_2, \tag{5.9}$$

де T_2 — ще одне задане значення порогу.

Якщо контраст є різко вираженим, то $\beta_+ \approx \alpha$ і $\beta_- \approx \alpha$. Якщо ж контраст поступовий, то інтервал $[\beta_+, \beta_-]$ являє собою його *перехідну зону*.

Класифікація пікселів $p(x, y)$ на дві зони визначається наступними правилами:

- для *поступового* та *виразного* контрастів:
  - якщо $p(x, y) \le F(\beta_-)$, піксель $p(x, y)$ належить до зони «1»;
  - якщо $p(x, y) \ge F(\beta_+)$, піксель $p(x, y)$ належить до зони «2»;
  - якщо ж $F(\beta_-) < p(x, y) < F(\beta_+)$, піксель $p(x, y)$ належить до *перехідної зони*;
- для *шумового* контрасту пікселі блока поділяються на дві зони однакової розмірності:
  - якщо $p(x, y) < F(N^2/2)$, піксель $p(x, y)$ належить до зони «1»;
  - якщо $p(x, y) > F(N^2/2)$, піксель $p(x, y)$ належить до зони «2».

У блоках 1-го і 2-го типів контрасту зони з різною яскравістю не обов'язково повинні розташовуватися впритул одна до одної і зовсім не обов'язково повинні містити рівну кількість пікселів. Більше того, деякі пікселі взагалі можуть не належати жодній з цих зон. У блоках 3-го типу контрасту класифікація є складнішою.

Поділ зон на категорії

Після поділу пікселів блока на зони необхідно передбачити вбудовування біта шляхом модифікації певних характеристик цих зон.

Нажаль, як зазначають автори [100], результат безпосереднього впливу на зони або недостатньо стійкий до завад, або ж не задовольняє одержуваними у підсумку візуальними спотвореннями первинного зображення.

Пошук у складі зони блока того пікселя, що був би найбільш оптимальним для модифікації, полягає у поділі самої зони на дві категорії: ***A*** і ***Z***. Для сортування пікселів по цим категоріям на блоки зображення накладаються спеціальні маски, причому вкрай бажаною є індивідуальність масок для кожного конкретного блока. Призначення масок полягає у забезпеченні секретності стеганошляху.

Приклади масок розбиття зон на категорії наведено на рис. 5.20. При цьому в [100] рекомендується користуватися більш складними комбінаціями і, до того ж, змінювати маску при переході до приховання кожного наступного біта повідомлення.

A	*A*	*A*	*A*	*Z*	*Z*	*Z*	*Z*
A	*A*	*A*	*A*	*Z*	*Z*	*Z*	*Z*
A	*A*	*A*	*A*	*Z*	*Z*	*Z*	*Z*
A	*A*	*A*	*A*	*Z*	*Z*	*Z*	*Z*
Z	*Z*	*Z*	*Z*	*A*	*A*	*A*	*A*
Z	*Z*	*Z*	*Z*	*A*	*A*	*A*	*A*
Z	*Z*	*Z*	*Z*	*A*	*A*	*A*	*A*
Z	*Z*	*Z*	*Z*	*A*	*A*	*A*	*A*

а)

A	*A*	*Z*	*Z*	*A*	*A*	*Z*	*Z*
A	*A*	*Z*	*Z*	*A*	*A*	*Z*	*Z*
Z	*Z*	*A*	*A*	*Z*	*Z*	*A*	*A*
Z	*Z*	*A*	*A*	*Z*	*Z*	*A*	*A*
A	*A*	*Z*	*Z*	*A*	*A*	*Z*	*Z*
A	*A*	*Z*	*Z*	*A*	*A*	*Z*	*Z*
Z	*Z*	*A*	*A*	*Z*	*Z*	*A*	*A*
Z	*Z*	*A*	*A*	*Z*	*Z*	*A*	*A*

б)

Рис. 5.20. Приклади використовуваних категоріальних масок розмірами 4×4 (*а*) і 2×2 (*б*)

Категорія, до якої буде віднесено той чи інший піксель, залежатиме від двох факторів:

- просторового розташування пікселя в масиві блока;
- номера зони, до якої цей піксель було віднесено.

Також важливо відзначити, що алгоритм формування масок має триматися у секреті, оскільки знання конфігурації останніх суттєво знижує стійкість стеганосистеми в цілому.

Правила вбудовування бітів повідомлення

За результатами виконання перших трьох етапів одержують чотири різні групи пікселів у визначених блоках: в залежності від зон («1» або «2») та категорій (***A*** і ***Z***). Слід відзначити, що існує ще й п'ята група

пікселів: ті, що не ввійшли до жодної із зон. Проте такі пікселі не беруть участі в подальшому аналізі.

Для зазначених чотирьох підмножин можуть бути обчислені шість параметрів:

- чотири середніх значення яскравостей λ_{1A}, λ_{1Z}, λ_{2A} і λ_{2Z} для груп, що містять відповідно n_{1A}, n_{1Z}, n_{2A} і n_{2Z} пікселів;
- два середніх значення яскравості відповідних зон: Λ_1 і Λ_2.

Середні значення яскравостей однакових зон об'єднуються і, таким чином, один біт повідомлення вбудовується до кожної з зон. Це підвищує стійкість системи, дозволяючи, разом з цим, вбудовувати біти без надмірної зміни блоків.

Вбудовування біта b до блока виконується у відповідності до зв'язків між категоріями середніх значень яскравості. Правило вбудовування наступне:

$$\begin{cases} \lambda^*_{1Z} - \lambda^*_{1A} = E; \\ \lambda^*_{2Z} - \lambda^*_{2A} = E, \end{cases} \text{ при } b = 0; \qquad (5.10, а)$$

$$\begin{cases} \lambda^*_{1A} - \lambda^*_{1Z} = E; \\ \lambda^*_{2A} - \lambda^*_{2Z} = E, \end{cases} \text{ при } b = 1, \qquad (5.10, б)$$

де λ^*_{1A}, λ^*_{1Z}, λ^*_{2A} і λ^*_{2Z} — середні значення яскравості, необхідні для приховання біта b; E — рівень (поріг) вбудовування, тобто необхідна різниця між зазначеними середніми значеннями.

Для того щоб зробити результат вбудовування якомога непомітнішим, у зображенні мають бути збережені низькі частоти (до яких, як відомо, є найбільш чутливою ЗСЛ). Збереження середніх значень інтенсивностей кожної зони забезпечується виконанням наступних умов:

$$\frac{n_{1A} \cdot \lambda^*_{1A} + n_{1Z} \cdot \lambda^*_{1Z}}{n_{1A} + n_{1B}} = \Lambda_1; \qquad (5.11, а)$$

$$\frac{n_{2A} \cdot \lambda^*_{2A} + n_{2Z} \cdot \lambda^*_{2Z}}{n_{2A} + n_{2B}} = \Lambda_2. \qquad (5.11, б)$$

У сукупності, формули (5.10) і (5.11) дозволяють визначити значення λ^*_{1A}, λ^*_{1Z}, λ^*_{2A} і λ^*_{2Z}.

Яскравість пікселів кожної зони має бути адаптованою для збереження значень Λ_1 і Λ_2. При цьому вважається, що зміна яскравості всіх пікселів, що належать до спільної зони, є однаковою.

Позначимо через Δ_{1A}, Δ_{1Z}, Δ_{2A} і Δ_{2Z} зміни яскравості пікселів відповідних груп. Тоді маємо:

$$\Delta_{ij} = \lambda^*_{ij} - \lambda_{ij}, \tag{5.12}$$

де $i = \{1, 2\}$; $j = \{A, Z\}$.

Видобування бітів повідомлення

Видобування прихованої інформації з контейнера вимагає наявності попередньої інформації про розмірність блоків, на які розбивалося зображення, а також про конфігурацію масок, що використовувалися при вбудовуванні. Процес видобування складається з наступних чотирьох етапів:

1) розбиття зображення на блоки розмірністю $N \times N$ (8×8);

2) класифікація пікселів окремого блока на зони;

3) поділ кожної зони на категорії;

4) співставлення середніх значень яскравості для визначення значення вбудованого біта даних.

Перші три етапи є ідентичними до відповідних етапів алгоритму вбудовування. А ось четвертий етап пропонується розглянути докладніше.

Нехай Σ_1 і Σ_2 — значення, одержані шляхом порівняння між собою середніх значень яскравості:

$$\Sigma_1 = \lambda_{1A} - \lambda_{1B}; \tag{5.13, а}$$

$$\Sigma_2 = \lambda_{2A} - \lambda_{2B}. \tag{5.13, б}$$

Знак обчислених Σ_1 і Σ_2 дозволяє зробити припущення щодо істинного значення прихованого біта. Крім того, абсолютні значення $|\Sigma_1|$ і $|\Sigma_2|$ несуть інформацію про рівень достовірності такого припущення. При цьому можливі три випадки:

1) $\Sigma_1 \cdot \Sigma_2 > 0$. Тоді $b^* = 1$, якщо $\Sigma_1 > 0$, і $b^* = 0$, якщо $\Sigma_1 < 0$. За таких умов рівень достовірності:

- *вельми високий*, якщо $|\Sigma_1|$ і $|\Sigma_2| > 2$;
- *досить високий*, якщо $|\Sigma_1|$ або $|\Sigma_2| > 2.5$;
- *низький*, якщо $|\Sigma_1|$ і $|\Sigma_2| < 0.7$;
- *високий* — в усіх інших випадках.

2) $\Sigma_1 \cdot \Sigma_2 < 0$. У такому разі додатково обчислюється наступний параметр:

$$\Sigma' = \Sigma_1 \cdot \left(n_{1A} + n_{1Z}\right) + \Sigma_2 \cdot \left(n_{2A} + n_{2Z}\right).$$

І тоді, якщо $\Sigma' > 0$, то $b^* = 1$; якщо $\Sigma' < 0$, $b^* = 0$; якщо $\Sigma' = 0$, $b^* = ?$. В усіх зазначених випадках рівень достовірності є *низьким*.

3) $\Sigma_1 \cdot \Sigma_2 \approx 0$. За таких обставин проводиться обчислення параметра

$$\Sigma' = max\left(|\Sigma_1|, |\Sigma_2|\right).$$

Тоді $b^* = 1$, якщо $\Sigma' > 0$; $b^* = 0$, якщо $\Sigma' < 0$; і $b^* = ?$, якщо $\Sigma' = 0$. Рівень достовірності при цьому є *низьким*.

Для підвищення загальної завадостійкості автори методу пропонують використовувати корегувальні коди на зразок БЧХ [65, 115].

Розглянемо реалізацію описаного методу у програмному середовищі *Mathcad*.

Крок 1

Виділимо масиви колірних компонентів зображення-контейнера:

R := **READ_RED**("C.bmp");
G := **READ_GREEN**("C.bmp")
B := **READ_BLUE**("C.bmp").

Аналогічно попереднім методам, вбудовування проводитимемо до масиву **B** синьої колірної складової.

Крок 2

Визначимо розмірність масиву контейнера **X** та **Y**, а також задамо розмірність сегментів (блоків) **N**, на які він розбиватиметься:

X := **rows**(**B**) = 128 пікс.; **Y** := **cols**(**B**) = 128 пікс.; **N** := 8 сегм.

При цьому загальна кількість сегментів:

$$\mathbf{N_S} := \mathbf{X} \cdot \mathbf{Y} / \mathbf{N}^2 = 256 \text{ сегм.}$$

Крок 3

За допомогою програмного модуля (М.34) проводимо розбиття загального масиву **B** на окремі блоки: останні виокремлюються за допомогою функції **submatrix**(...) і зберігаються як елементи вектора **Б**.

На початковому етапі, стовпці, що обмежують виокремлюваний з масиву сегмент, мають індекси **c1** ← 1, **c2** ← **N**. Поступово нарощуючи значення індексів рядків **r1** і **r2**, повністю проходиться ділянка, обмежена стовпцями **c1** і **c2**. У випадку, коли індекс **r2** збігається з індексом останнього рядка (**X**), індекси стовпців збільшуються на ширину виокремлюваних сегментів **N** і процес повторюється, допоки не буде видобутий останній, $\mathbf{N_S}$-й сегмент.

```
Б := | c1 ← 1 , c2 ← N
     | for b ∈ 1.. N_S
     |   | r1 ← mod[N·(b − 1) + 1 , X]
     |   | r2 ← r1 + N − 1
     |   | Б_b ← submatrix(B , r1 , r2 , c1 , c2)
     |   | if r2 = X
     |   |   | c1 ← c1 + N
     |   |   | c2 ← c2 + N
     | Б
```
(М.34)

Зауважимо, що даний модуль не розрахований на розбиття зображення на сегменти, розмірність яких не є кратною розмірності масиву зображення. Зазначений випадок можна врахувати або обрізкою зображення, або введенням до тіла програмного модуля (М.34) відповідних обмежуючих умов. У нашому випадку, за результатом обчислення (М.34) повертається вектор **Б** з 256 елементів, кожен з яких являє собою матрицю 8×8.

Окремий елемент зі складу **Б** призначено для приховування одного біта секретного повідомлення **М**. Тому перед вбудовуванням бажано перевірити достатність кількості сегментів для виконання цієї операції. Повідомлення є ідентичним використовуваному при розгляді попередніх методів; його довжина $L_M = 176$ бітів. Таким чином, кількість сегментів розмірністю 8×8, отриманих з масиву 128×128, є цілком достатньою ($176 < 256$).

Крок 4

Проводимо класифікацію пікселів кожного з L_M сегментів на зони «0» (перехідна зона), «1» і «2» (М.35). На початку циклу зміни **b** змінній **B** (велика бета) присвоюється значення **b**-го сегмента **N**×**N** зі складу масиву **Б**. Поточний масив **B** розгортається у вектор **f**, який у відсортованому вигляді присвоюється змінній **F**. Таким чином, вектор **F** налічує N^2 елементів, що йдуть за зростанням.

Для існування можливості прийняття однакових рішень як на передавальній стороні при вбудовуванні, так і на приймальній при видобуванні стосовно точки, в якій функція **F** має найбільшу крутизну (див. рис. 5.19), останню бажано згладити, використовуючи в якості вузлових точок лише невелику частину від загальної їх кількості N^2 (наприклад, вісім: $x \in 1..8$). Зрозуміло, що абсциса першої вузлової точки має дорівнювати одиниці ($L_1 = 1$), останньої — N^2 ($L_8 = 64$). Абсциси проміжних вузлових точок формуються за виразом, зміст якого є цілком очевидним. Відповідні ординати вузлових точок заносяться до масиву **ф**.

```
Zone := | for b ∈ 1.. L_M                                                    (M.35)
        |   B ← Б_b
        |   for k ∈ 1.. N
        |     f ← if(k = 1, B^⟨1⟩, stack(f, B^⟨k⟩))
        |   F ← sort(f)
        |   for x ∈ 1.. 8
        |     | L_x ← (x − 1)·round(N^2 ÷ 7) + 1
        |     | ϕ_x ← F_{L_x}
        |   spline ← lspline(L, ϕ)
        |   S_max ← 0 , α ← 0
        |   for ω ∈ 1.. N^2
        |     if d/dω interp(spline, L, ϕ, ω) > S_max
        |       | S_max ← d/dω interp(spline, L, ϕ, ω)
        |       | α ← ω
        |   α ← if(α = 0, N^2/2, if(α = 1, 2, if(α = N^2, N^2 − 1, α)))
        |   zone ← matrix(N, N, f(x, y) ← mod(x+y, 2)+1) if S_max < T_1
        |   if S_max ≥ T_1
        |     for λ ∈ α.. 1
        |       | β_ ← α if λ = 1
        |       | β_ ← λ , break if (F_α − F_λ) > T_2
        |     for λ ∈ α.. N^2
        |       | β+ ← α if λ = N^2
        |       | β+ ← λ , break if (F_λ − F_α) > T_2
        |     for x ∈ 1.. N
        |       for y ∈ 1.. N
        |         | zone_{x,y} ← 1 if B_{x,y} ≤ F_{β_}
        |         | zone_{x,y} ← 2 if B_{x,y} ≥ F_{β+}
        |         | zone_{x,y} ← 0 otherwise
        |   Zone_b ← zone
        | Zone
```

Користуючись вбудованою функцією *Mathcad* **lspline**(**L**, **ϕ**), формуємо вектор кубічних сплайн-коефіцієнтів других похідних при наближенні до опорних точок (при цьому граничні точки є лінійними). Для проведення сплайн-апроксимації використовуємо іншу вбудовану функцію — **interp**(**spline**, **L**, **ϕ**, **ω**), яка для кожної шуканої абсциси **ω** обчислює значення апроксимованої функції.

З основ диференціального числення відомо [101], що кут нахилу дотичної в заданій точці до графіку функції дорівнює арктангенсу від похідної цієї функції при даному значенні аргументу. Отже, абсцису точки α максимальної крутизни сплайн-апроксимованої функції ϕ можна знайти за максимумом першої похідної $(d/d\omega)$ даної функції серед усіх можливих значень ω. У випадку відшукання максимального значення крутизни, змінній α присвоюється відповідне значення ω.

Якщо для будь-якого ω не знайдено похідної, більшої за S_{max} (усі пікселі сегмента мають однакову інтенсивність), то змінній α присвоюється значення $N^2/2$. Якщо точка максимальної крутизни є граничною (тобто дорівнює 1 або N^2), то вона зсувається на один крок праворуч або ліворуч відповідно (за недотримання такої поправки пікселі, що належать, відповідно, до зони «1» або «2», будуть відсутніми).

У тому випадку, коли одержане значення крутизни виявляється меншим за поріг T_1, пікселі сегмента діляться між зонами «1» і «2» порівну у шаховому порядку. Якщо ж $S_{max} \geq T_1$, проводиться пошук абсцис β_- та β_+.

При значенні інтенсивності пікселя $B_{x,y} \leq F_{\beta_-}$ останній відноситься до зони «1»; при $B_{x,y} \geq F_{\beta_+}$ — до зони «2»; при $F_{\beta_-} < B_{x,y} < F_{\beta_+}$ — до перехідної зони («0»).

За результатом обчислення (М.35) одержується вектор **Zone**, кожен елемент якого являє собою матрицю ідентичної розмірності із відповідним сегментом зображення ($N \times N$). У свою чергу, **b**-й елемент такої матриці може приймати три значення: 1, 2 або 0, що, зрештою, і визначає, до якої з можливих зон належить піксель сегмента $Б_b$.

Приклад результатів поділу зображення на сегменти та класифікації пікселів деяких з цих сегментів на зони представлені на рис. 5.21 (нумерація сегментів наступна: у першому стовпці — з 1-го по 16-й, у другому — з 17-го по 32-й і т. д.; встановлені значення порогів: $T_1 := 6$, $T_2 := 3$).

Крок 5

Проводимо розбиття кожної зони на категорії "A" і "Z" у відповідності до індивідуальних масок. Простий програмний модуль генерування останніх наведено нижче — (М.36).

Для кожного з N_S сегментів зображення генерується масив маски μ_i розмірністю $N \times N$. Причому на початку генерування створюється N^2-елементний вектор μ' з самих лише категорій "Z", половина з яких з псевдовипадковим кроком замінюється значенням "A".

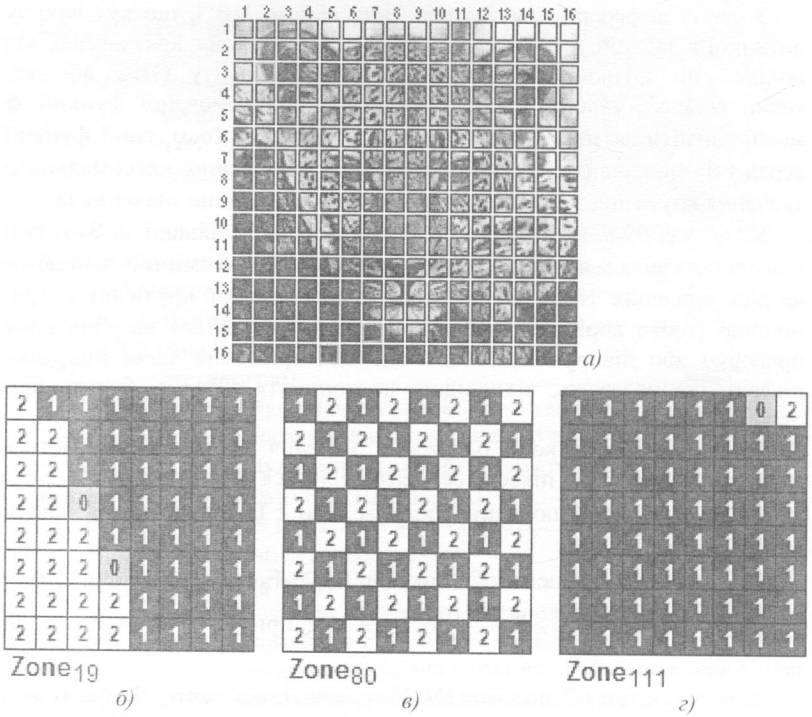


Рис. 5.21. Схема поділу зображення на сегменти (*а*) і приклади класифікації пікселів 19-го (*б*), 80-го (*в*) і 111-го (*г*) сегментів по зонам

$$\mu := \left| \begin{array}{l} \text{for } i \in 1..N_S \\ \quad \text{for } j \in 1..N^2 \\ \quad\quad \mu'_j \leftarrow \text{"Z"} \\ \quad \text{for } j \in 1..N^2 \div 2 \\ \quad\quad \mu'_{\text{mod}\left(i+j\cdot K_o, N^2\right)+1} \leftarrow \text{"A"} \\ \quad \mu_i \leftarrow \text{submatrix}(\mu', 1, N, 1, 1) \\ \quad \text{for } c \in 2..N \\ \quad\quad \mu_i \leftarrow \text{augment}\left[\mu_i, \text{submatrix}[\mu', c\cdot N-(N-1), c\cdot N, 1, 1]\right] \\ \mu \end{array} \right. \qquad \text{(M.36)}$$

Індекси категорій "A" визначаються заздалегідь обумовленим значенням параметра K_o, а також поточними значенням змінних циклів **i** та **j**. Для того, щоб не вийти за межі розмірності вектора, використо-

вується функція повернення залишку від ділення **mod**(...). Після заповнення, вектор **μ'** згортається до матриці μ_i.

Приклади деяких масок, одержаних для $K_o := 123$, зображені на рис. 5.22.

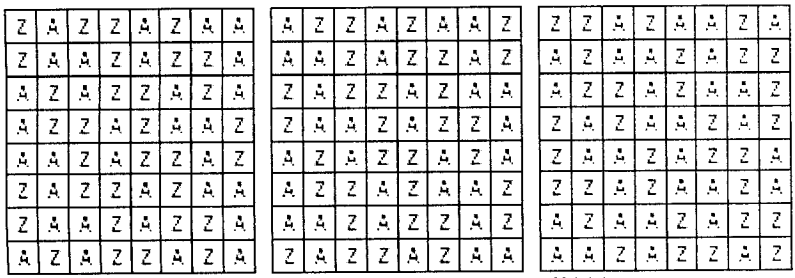


Рис. 5.22. Приклади масок розбиття по категоріям для сегментів 19, 80 і 111

Крок 6

Програмний модуль вбудовування бітів (М.37) побудований на основі виразів (5.10)–(5.12).

На початку модуля на основі приховуваного повідомлення **M** створюється вектор двійкових даних. З урахуванням відповідних масивів зон $Z \leftarrow Zone_b$ та маски $м \leftarrow \mu_b$, циклом зміни **b** проводиться вбудовування окремого біта $bit \leftarrow M_{vec\_bin_b}$ до блока $B \leftarrow Б_b$.

Розглянемо процедуру вбудовування більш детально. На основі даних щодо належності пікселів **b**-го сегмента до зони «1» або «2», а також до категорії "A" чи "Z", проводиться підрахунок кількості пікселів, що задовольняють одній з можливих комбінацій зон і категорій (1A, 1Z, 2A, 2Z). Результати заносяться до відповідного елемента масиву **n** (рис. 5.23), який на початку підрахунку вбудованою функцією **matrix**(...) заповнюється нулями.

	1	2	
1	1A	1Z	*зони*
2	2A	2Z	
	категорії		

Рис. 5.23. Конфігурація масивів **n** та **Σλ**

Аналогічно проводиться підрахунок загальної яскравості $\Sigma\lambda$ пікселів, що належать тій чи іншій парі зон і категорій (конфігурація заповнюваного при цьому масиву відповідає попередньому випадку — див. рис. 5.23).

За одержаними результатами обчислюється масив λ середніх значень яскравості кожної з чотирьох груп пікселів. Для скорочення програмних рядків дане обчислення ведеться з використанням операції векторизації — одночасного проведення деякої скалярної математичної операції (у нашому випадку — ділення) над усіма елементами масиву або масивів, поміченими знаком векторизації «$\overrightarrow{\ }$». Звичайно ж, така паралельність обчислення відноситься не до самих обчислень, а лише до їх алгоритмічного запису. Тому кардинального зменшення часу виконання операції чекати при цьому не доводиться.

$\Delta :=$ | Ⓘ — див. (М.21), крім $S \leftarrow C$ (М.37)

for $b \in 1..L_M$

$bit \leftarrow M_{vec\_bin_b}$, $B \leftarrow Б_b$, $Z \leftarrow Zone_b$, $м \leftarrow \mu_b$

$n \leftarrow matrix(2,2,f(i,j) \leftarrow 0)$, $\Sigma\lambda \leftarrow n$

for $x \in 1..N$

for $y \in 1..N$

$i \leftarrow Z_{x,y}$, $j \leftarrow if(м_{x,y} = "A", 1, 2)$

if i

$n_{i,j} \leftarrow n_{i,j} + 1$

$\Sigma\lambda_{i,j} \leftarrow \Sigma\lambda_{i,j} + B_{x,y}$

$\lambda \leftarrow \overrightarrow{\left(\frac{\Sigma\lambda}{n}\right)}$

for $i \in 1..2$

$\Lambda_i \leftarrow \frac{\lambda_{i,1} \cdot n_{i,1} + \lambda_{i,2} \cdot n_{i,2}}{n_{i,1} + n_{i,2}}$

$\lambda'_i \leftarrow lsolve\left[\begin{pmatrix} n_{i,1} & n_{i,2} \\ 2 \cdot bit - 1 & 1 - 2 \cdot bit \end{pmatrix}, \begin{bmatrix} \Lambda_i \cdot (n_{i,1} + n_{i,2}) \\ E \end{bmatrix}\right]$

$\lambda^* \leftarrow augment(\lambda'_1, \lambda'_2)^T$

$\Delta_b \leftarrow \lambda^* - \lambda$

Δ

Далі у (М.37) для кожної зони проводиться розрахунок середніх значень яскравості Λ (див. формули (5.11)), що мають бути збережені й після проведення вбудовування біта до сегмента зображення.

Для розв'язку системи рівнянь {(5.10); (5.11)} з двома невідомими ($\lambda^*_{1,1}$ і λ^*_{1Z} або $\lambda^*_{2,1}$ і λ^*_{2Z}) використано вбудовану функцію *Mathcad* розв'язку лінійної системи з ***n*** рівнянь при ***n*** невідомих — **lsolve**(*H*,*V*), де *H* — квадратна несингулярна матриця; *V* — вектор, що має ту саму кількість рядків, що й матриця *H*.

Для зручності подальших пояснень, перепишемо дані системи у відповідності до прийнятих у програмному модулі (М.37) позначень:

- при **bit** = 1:

$$\left.\begin{array}{c} \mathbf{n}_{i,1}\cdot\boldsymbol{\lambda}_{i,1}+\mathbf{n}_{i,2}\cdot\boldsymbol{\lambda}_{i,2}=\boldsymbol{\Lambda}_i\cdot\left(\mathbf{n}_{i,1}+\mathbf{n}_{i,2}\right), \\ \boldsymbol{\lambda}_{i,1}-\boldsymbol{\lambda}_{i,2}=\mathbf{E}; \end{array}\right\} \Rightarrow$$

$$\Rightarrow\ H=\begin{pmatrix} \mathbf{n}_{i,1} & \mathbf{n}_{i,2} \\ 1 & -1 \end{pmatrix},\quad V=\begin{pmatrix} \boldsymbol{\Lambda}_i\cdot\left(\mathbf{n}_{i,1}+\mathbf{n}_{i,2}\right) \\ \mathbf{E} \end{pmatrix};$$

- при **bit** = 0:

$$\left.\begin{array}{c} \mathbf{n}_{i,1}\cdot\boldsymbol{\lambda}_{i,1}+\mathbf{n}_{i,2}\cdot\boldsymbol{\lambda}_{i,2}=\boldsymbol{\Lambda}_i\cdot\left(\mathbf{n}_{i,1}+\mathbf{n}_{i,2}\right), \\ \boldsymbol{\lambda}_{i,2}-\boldsymbol{\lambda}_{i,1}=\mathbf{E}; \end{array}\right\} \Rightarrow$$

$$\Rightarrow\ H=\begin{pmatrix} \mathbf{n}_{i,1} & \mathbf{n}_{i,2} \\ -1 & 1 \end{pmatrix},\quad V=\begin{pmatrix} \boldsymbol{\Lambda}_i\cdot\left(\mathbf{n}_{i,1}+\mathbf{n}_{i,2}\right) \\ \mathbf{E} \end{pmatrix},$$

де **i** — номер зони.

За будь-якого значення **bit**, функція **lsolve**(...) обчислюється двічі — для кожної із зон. Результатом кожного з цих обчислень є вектор $\boldsymbol{\lambda}'_i$ з двох елементів, значення яких відповідають шуканим невідомим. Одержані у підсумку два вектори об'єднуємо стовпець до стовпця, а результат об'єднання транспонуємо (щоб перейти до конфігурації, зображеній на рис. 5.23).

На завершальному етапі циклу зміни **b** обчислюється різниця (5.12). Результат обчислення (який, зрозуміло, також має конфігурацію, що відповідає рис. 5.23) присвоюється **b**-му елементу вектора **Δ**. Приклади отриманих результатів наведені на рис. 5.24.

-18,547	21,475
-22,971	20,883

Δ_{19}

-17,375	17,375
-16,094	16,094

Δ_{80}

18,452	-18,452
0,000	74,000

Δ_{111}

Рис. 5.24. Приклади результатів обчислень матриці зміни яскравості Δ_b

Крок 7

Використовуючи програмний модуль (М.38), проводимо стеганографічну модифікацію сегментів зображення ($Б_b \rightarrow БS_b$). При цьому, якщо номер блока **b** перевищує загальну кількість бітів у повідомленні (L_M), яскравість пікселів у блоці залишається без змін.

В якості прикладу на рис. 5.25 наведено результати змін яскравостей пікселів сегмента № 19 за умови наведених на рис. 5.21 і 5.22 класифікації останніх по зонам і маски розбиття по категоріям.

$$
\begin{aligned}
&БS := \Big| \text{for } b \in 1..N_S \\
&\quad \Big| B \leftarrow Б_b \\
&\quad \Big| \text{if } b \le L_M \\
&\quad \Big| \text{for } x \in 1..N \\
&\quad \Big| \quad \text{for } y \in 1..N \\
&\quad \Big| \quad \Big| i \leftarrow (Zone_b)_{x,y},\ j \leftarrow \text{if}\left[(\mu_b)_{x,y} = \text{"A"}, 1, 2\right] \\
&\quad \Big| \quad \Big| B_{x,y} \leftarrow B_{x,y} + (\Delta_b)_{i,j} \quad \text{if } i \\
&\quad \Big| БS_b \leftarrow B \\
&\quad БS
\end{aligned}
\tag{М.38}
$$

20,883	-18,547	21,475	21,475	-18,547	21,475	-18,547	-18,547
20,883	-22,971	-18,547	21,475	-18,547	21,475	21,475	-18,547
-22,971	20,883	-18,547	21,475	21,475	-18,547	21,475	-18,547
-22,971	20,883	0,000	-18,547	21,475	-18,547	-18,547	21,475
-22,971	-22,971	20,883	-18,547	21,475	21,475	-18,547	21,475
20,883	-22,971	20,883	0,000	-18,547	21,475	-18,547	-18,547
20,883	-22,971	-22,971	20,883	-18,547	21,475	21,475	-18,547

Рис. 5.25. Приклад зміни яскравості пікселів 19-го сегмента: $БS_{19} - Б_{19}$

Крок 8

Повертаємо блоки $БS_b$ на відповідні їм місця у графічному контейнері, користуючись модулем (М.39). На першому етапі перші **X/N** блоків (у нашому випадку — 16) групуються в один спільний масив **Cm** шляхом об'єднання останнього рядка попереднього блока з першим рядком наступного (функція **stack**(...)). У подальшому, після об'єднання в такий спосіб **X/N** блоків, результуючий масив **Cm** приєднується до підсумкового масиву **S** — стовпець до стовпця (функція **augment**(...)).

За цим масив **Cm** знищується і виконується формування наступної групи з **X / N** блоків.

$$
\begin{array}{l}
S := \left| \begin{array}{l}
Cm \leftarrow 0 \\
\text{for } b \in 1..N_S \\
\quad \left| \begin{array}{l}
Cm \leftarrow \text{if}(Cm = 0, БS_b, \text{stack}(Cm, БS_b)) \\
\text{if } \neg \text{mod}\left(b, \frac{X}{N}\right) \\
\quad \left| \begin{array}{l}
S \leftarrow \text{if}\left(b = \frac{X}{N}, Cm, \text{augment}(S, Cm)\right) \\
Cm \leftarrow 0
\end{array} \right. \\
\end{array} \right. \\
S
\end{array} \right.
\end{array}
\qquad \text{(M.39)}
$$

Відтворене на основі масиву **S** зображення (у нашому випадку — канал синього) у більшості випадків буде занадто спотвореним через вже згаданий вище вихід значень інтенсивностей пікселів за межі [0; 255] — рис. 5.26. Тому його необхідно додатково пронормувати, скориставшись, наприклад, вбудованою функцією **scale**(...) (див., наприклад, підпункт 5.3.2.7):

$$S_{norm} := \text{scale}(S, 0, 255).$$

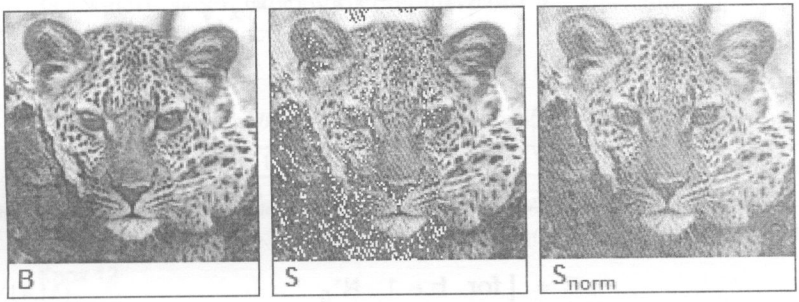


Рис. 5.26. Порівняння первинного (**B**), модифікованого (**S**) (при значенні **E** = 35) і додатково нормованого (S_{norm}) зображень-контейнерів

Крок 9

Перейдемо до видобування прихованого повідомлення. Прийняті позначення: **S*** — заповнений контейнер; **X*** та **Y*** — розміри контейнера по вертикалі (кількість рядків пікселів) та горизонталі (кількість стовпців) відповідно; **N*** — розмірність виокремлюваного блока

контейнера; N^*_S — загальна кількість блоків; T^*_1, T^*_2 — значення першого і другого порогів порівняння; μ^* — масив секретних масок.

Програмні модулі поділу зображення на блоки (**Б***), класифікації пікселів блока на зони (**Zone***) є аналогічними, відповідно, модулям (М.34) і (М.35).

Обчислення кількості пікселів **n**, що задовольняють одній з можливих комбінацій зон і категорій, а також середніх значень яскравості λ кожної з чотирьох груп пікселів виконується у програмному модулі (М.40).

$$\begin{pmatrix} n^* \\ \lambda^* \end{pmatrix} := \left| \begin{array}{l} \text{for } b \in 1..N^*_S \\ \quad n \leftarrow \text{matrix}(2,2,f(i,j) \leftarrow 0),\ \Sigma\lambda \leftarrow n \\ \quad \text{for } x \in 1..N^* \\ \quad\ \text{for } y \in 1..N^* \\ \quad\quad i \leftarrow (Zone^*_b)_{x,y} \\ \quad\quad j \leftarrow \text{if}\left[(\mu^*_b)_{x,y} = \text{"A"}, 1, 2\right] \\ \quad\quad \text{if } i \\ \quad\quad\quad n_{i,j} \leftarrow n_{i,j} + 1 \\ \quad\quad\quad \Sigma\lambda_{i,j} \leftarrow \Sigma\lambda_{i,j} + (Б^*_b)_{x,y} \\ \quad n^*_b \leftarrow n,\ \lambda^*_b \leftarrow \overrightarrow{\left(\frac{\Sigma\lambda}{n}\right)} \\ \begin{pmatrix} n^* \\ \lambda^* \end{pmatrix} \end{array} \right. \tag{М.40}$$

Порівняння середніх значень яскравості для кожної з зон (формула (5.13)) проводимо, використовуючи модуль (М.41).

$$\Sigma := \left| \begin{array}{l} \text{for } b \in 1..N^*_S \\ \quad \text{for } i \in 1..2 \\ \quad\quad E_i \leftarrow (\lambda^*_b)_{i,1} - (\lambda^*_b)_{i,2} \\ \quad \Sigma_b \leftarrow E \\ \Sigma \end{array} \right. \tag{М.41}$$

Безпосередньо модуль видобування — (М.42), в якому реалізовано перевірку умов, викладених у теоретичному описі даного методу. Вектор двійкових даних $M^*_{vec\_bin}$, що повертається модулем (М.42), перетворюється на рядок символів **M*** аналогічно тому, як це робилося при моделюванні попередніх методів.

$$
\begin{aligned}
&M^*_{vec\_bin} := \Big| \text{for } b \in 1 .. N^*_S \\
&\quad \Big| E \leftarrow \Sigma_b,\ n \leftarrow n^*_b \\
&\quad \Big| M^*_b \leftarrow E_1 > 0 \text{ if } E_1 \cdot E_2 > 0 \\
&\quad \Big| \text{if } E_1 \cdot E_2 < 0 \\
&\quad \Big| \quad \Big| E' \leftarrow E_1 \cdot (n_{1,1} + n_{1,2}) + E_2 \cdot (n_{2,1} + n_{2,2}) \\
&\quad \Big| \quad \Big| M^*_b \leftarrow E' > 0 \\
&\quad \Big| \text{otherwise} \\
&\quad \Big| \quad \Big| E'' \leftarrow \max(|E_1|, |E_2|) \\
&\quad \Big| \quad \Big| M^*_b \leftarrow E'' > 0 \\
&M^*
\end{aligned}
\qquad \text{(M.42)}
$$

У нашому випадку був отриманий наступний текстовий рядок:

$$M^* = \text{"© Alex Puzyrenko, 2017Úê+IYáRÝ#"}$$

Крок 10

Одержані при обчисленні візуального спотворення результати зведено до табл. 5.6 (див. стор. 164).

Крок 11

Залежність якості видобутої стеганограми від значення порогу **E** (при $T_1 = 6$ і $T_2 = 3$) простежена у табл. 5.4.

Таблиця 5.4

Залежність *BER* від величини порогу вбудовування

E, *ум.рівн.*	0.01	0.5	1	2	3	4	5	6	7	8	9	10
BER, %	15.34	13.64	13.64	12.5	7.96	5.11	6.25	3.98	2.27	0.57	0.0	0.0

Крок 12

Результат впливу стиснення *JPEG* на можливість видобування прихованих даних за різних значень порогу **E** наведено у табл. 5.5.

Таблиця 5.5

Залежність *BER* від рівня збереженої якості при *JPEG* стисненні

Якість, %		100	95	90	85	80	75	65	50	30	0
BER, %	**E** = 10	0.0	0.57	3.41	13.64	31.25	38.07	41.48	43.75	44.32	[illegible]
	E = 35	0.0	0.0	1.14	2.84	3.41	5.11	8.52	23.87	34.09	41.48

5.3.2.9. *Інші методи стеганографічного приховування у просторовій області зображення*

Дещо специфічним є метод, заснований на копіюванні блоків з однієї випадково обраної текстурної області до іншої, що має схожі статистичні характеристики [14]. Це призводить до появи в зображенні повністю ідентичних блоків. Виявлення останніх при цьому може бути проведено у такий спосіб:

- аналіз функції автокореляції стеганозображення і знаходження її піків;
- зсув масиву зображення відносно цих піків і віднімання отриманого зображення від його зсунутої копії;
- різниця в місцях розміщення копійованих блоків повинна бути близькою до нуля — це, в принципі, дає підстави скористатися певним порогом, а значення, менші за нього по абсолютній величині, вважати шуканими блоками.

Оскільки копії блоків є ідентичними, очевидно, що за тих або інших перетворень усього зображення вони змінюватимуться однаково. Якщо зробити розмір блоків достатньо великим, наведений метод характеризуватиметься стійкістю до більшості з негеометричних спотворень: у проведених авторами [14] експериментах була підтверджена стійкість до фільтрації зображення, стиснення його із втратами, обертання тощо.

Основним недоліком такого підходу до стеганографічного приховання даних бачиться виняткова складність знаходження у зображенні достатньої кількості областей, блоки з яких могли б бути замінені без помітного погіршення якості останнього. А це, у свою чергу, висуває недвозначні умови щодо достатньої текстурованості графічних контейнерів.

Алгоритм, запропонований у [102], дозволяє вбудовувати дані до блоків зображення-контейнера, що мають розмірність 8×8. На початку алгоритму створюється маска $\mu(x, y)$, розмірність якої відповідає розмірності масиву контейнера, а елементами є псевдовипадково розподілені 0 та 1: $\mu(x, y) \in \{0; 1\}$. Кожен блок $\boldsymbol{Б}$ в залежності від значення елементів маски ділиться на два підмасиви $\boldsymbol{Б}_1$ і $\boldsymbol{Б}_2$, для кожного з яких обчислюються середні значення яскравості — λ_1 і λ_2. Біт приховуваного повідомлення вбудовується за наступним правилом:

$$s(x, y) = \begin{cases} 1, \text{ при } \lambda_1 - \lambda_2 > E; \\ 0, \text{ при } \lambda_1 - \lambda_2 < -E, \end{cases} \tag{5.14}$$

де E — деякий поріг (необхідна для розрізнення бітів різниця між зазначеними середніми значеннями яскравості).

У тих випадках, коли умова (5.14) не виконується, відповідним чином змінюють значення яскравості пікселів одного з підмасивів ($\boldsymbol{Б}_1$ або $\boldsymbol{Б}_2$).

Для видобування біта прихованого повідомлення проводяться обчислення відповідних середніх значень яскравості підмасивів — λ^*_1 і λ^*_2. Різниця між ними дозволяє визначити значення прихованого біта:

$$b_i = \begin{cases} 1, & \text{при } \lambda^*_1 - \lambda^*_2 > 0; \\ 0, & \text{при } \lambda^*_1 - \lambda^*_2 < 0; \\ ?, & \text{при } \lambda^*_1 - \lambda^*_2 = 0. \end{cases} \qquad (5.15)$$

5.3.3. Приховування даних у частотній області зображення

Як вже зазначалося вище, стеганографічні методи приховування даних у просторовій області зображення є нестійкими до переважної більшості відомих видів спотворювань. Так, наприклад, застосування операції стиснення із втратами (стосовно зображення, зокрема, це може бути *JPEG*-компресія) призводить до часткового або, що більш імовірно, повного знищення вбудованої до контейнера інформації. Більш стійкими до різноманітних спотворювань, у тому числі й стиснення, є методи, що для приховування даних використовують не просторову область контейнера, а частотну.

Існує декілька способів декомпозиції зображення-контейнера і представлення його в частотній області. Для цього, наприклад, існують методи на основі використання дискретного косинусного перетворення (ДКП), дискретного перетворення Фур'є (ДПФ), вейвлет-перетворення, перетворення Карунена-Лоева та деякі інші. Подібні перетворення можуть застосовуватися або до окремих частин зображення, або ж до зображення в цілому.

У стеганографії серед усіх ортогональних перетворень найбільшого поширення зазнали дискретні косинусні і вейвлет-перетворення [5], що певною мірою пояснюється значним поширенням їх застосування під час стиснення зображень. Крім того, під час стеганографічного приховування даних є доцільним застосовувати саме те перетворення зображення, якому останнє згодом піддаватиметься при можливій атаці стисканням із втратами. Стеганоалгоритм може бути досить стійким до подальшого стиснення зображення лише у тому випадку, якщо він враховуватиме особливості алгоритму перспективного стиснення. Так, відомо, що алгоритм ДКП є базовим у форматі стиснення *JPEG*, а вейвлет-перетворення — у форматі *JPEG2000*. При цьому, звичайно,

Таблиця 5.6

Показники візуального спотворення у випадку приховування даних у просторовій області зображення

Назва показника спотворення	Оригінал	Методи приховування у просторовій області								
		заміна НЗБ, «*a*»	заміна НЗБ, «*б*»	НВ інтервал	НВП	блокове кодування	заміна палітри	квантування	Куттера-Джордана	Дармстелтера-Делейгла
Кількість біт у повідомленні, L_0	---	44328	56+176·56	176·56	176	176	176	176	25·176	176
Кількість модифікованих пікселів	0	8201	146	125	88	94	107	129	4276	[illegible]
Максимальна абсолютна різниця, *MD*	0	[illegible]	[illegible]	[illegible]	1	[illegible]	2	4	38	[illegible]
Середня абсолютна різниця, *AD*	0	0.501	$8.911\cdot10^{-3}$	$7.629\cdot10^{-3}$	$5.371\cdot10^{-3}$	$5.737\cdot10^{-3}$	$6.531\cdot10^{-3}$	$7.874\cdot10^{-3}$	4.78	[illegible]
Нормована середня абсолютна різниця, *NAD*	0	$5.697\cdot10^{-3}$	$1.014\cdot10^{-4}$	$8.684\cdot10^{-5}$	$6.114\cdot10^{-5}$	$6.53\cdot10^{-5}$	$7.434\cdot10^{-5}$	$8.962\cdot10^{-5}$	0.054	0.131
Середньоквадратична помилка, *MSE*	0	0.501	$8.911\cdot10^{-3}$	$7.629\cdot10^{-3}$	$5.371\cdot10^{-3}$	$5.737\cdot10^{-3}$	$8.606\cdot10^{-5}$	0.016	114.995	223.499
Нормована середньоквадратична помилка, *NMSE*	0	$4.336\cdot10^{-5}$	$7.719\cdot10^{-7}$	$6.608\cdot10^{-7}$	$4.652\cdot10^{-7}$	$4.97\cdot10^{-7}$	$7.454\cdot10^{-7}$	$1.348\cdot10^{-6}$	$9.961\cdot10^{-3}$	0.019
L^p-норма, $p = 2$	0	0.707	0.094	0.087	0.073	0.076	0.093	0.125	10.724	14.95
Лапласова середньоквадратична помилка, *LMSE*	0	$1.578\cdot10^{-3}$	$2.87\cdot10^{-5}$	$3.34\cdot10^{-5}$	$1.648\cdot10^{-5}$	$1.722\cdot10^{-5}$	$1.595\cdot10^{-5}$	$4.756\cdot10^{-5}$	0.337	0.75
Відношення «сигнал/шум», *SNR*	∞	$2.306\cdot10^{4}$	$1.296\cdot10^{6}$	$1.513\cdot10^{6}$	$2.149\cdot10^{6}$	$2.012\cdot10^{6}$	$1.342\cdot10^{6}$	$7.418\cdot10^{5}$	100.396	57.656
Макс. відношення «сигнал/шум», *PSNR*	∞	$1.299\cdot10^{5}$	$7.297\cdot10^{6}$	$8.523\cdot10^{6}$	$1.211\cdot10^{7}$	$1.133\cdot10^{7}$	$7.556\cdot10^{6}$	$4.178\cdot10^{6}$	565.46	[illegible]
Якість зображення, *IF*	1	0.999957	0.999999	0.999999	≈1	≈1	0.999999	0.999999	0.990039	0.986641
Нормована взаємна кореляція, *NC*	1	0.999539	0.999999	0.999991	1.000003	0.999989	0.999968	1.000016	[illegible]	0.997888
Якість кореляції, *CQ*	131.41	131.35	[illegible]	131.409	131.11	131.409	131.406	131.412	130.084	131.132
Структурний зміст, *SC*	1	1.000879	[illegible]	1.000016	0.999994	1.000022	1.000063	0.999966	1.010327	[illegible]
Загальне сигма-відношення «сигнал/шум», *GSSNR*	∞	$1.733\cdot10^{5}$	$7.363\cdot10^{6}$	$1.292\cdot10^{7}$	$1.513\cdot10^{7}$	$1.602\cdot10^{7}$	$1.02\cdot10^{7}$	$6.662\cdot10^{6}$	288.023	[illegible]
Сигма-відношення «сигнал/шум», *SSNR*	∞	222.2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	117.7	[illegible]
Нормоване відношення «сигма/помилка», *NSER*	256	56.571	240	160	182	183	240	186	23.762	80.0
Подібність гістограм, *HS*	0	2474	140	150	128	140	148	122	1240	[illegible]

стеганоалгоритм, до основи якого закладено вейвлет-перетворення, зовсім не обов'язково виявиться стійким до дискретного косинусного алгоритму стиснення, і навпаки. Автори [5] зазначають, що ще більш значні труднощі виникають при обранні методу стеганоперетворення під час приховання даних у потоковому відео. Причиною цього є те, що алгоритми стиснення відеоінформації (на додачу до стиснення нерухомого кадру) однією зі своїх обов'язкових складових мають ще й кодер векторів компенсування руху (при обробці нерухомих зображень таке компенсування є відсутнім через непотрібність). Для того, щоб бути в достатній мірі стійким, стеганоалгоритм має враховувати цей нюанс.

І досі залишається відкритим і питання існування стійкого стеганоперетворення, яке було б незалежним від застосовуваного у подальшому алгоритму стиснення. Розгляд різноманітних ортонормованих перетворень, таких як ДПФ, ДКП, перетворення Хартлі, субсмугове перетворення тощо з аспектів теорії інформації проведені, наприклад, авторами [19, 44].

Разом з тим, на сьогоднішній день відомо багато моделей для оцінки прихованої пропускної здатності стеганографічного каналу, орієнтованого на стійкість до атаки стисненням із втратами. Нижче наведено модель, представлену у роботах [5, 84].

Нехай, як і раніше, $\mathbf{C}$ — первинне зображення (контейнер-оригінал), $\mathbf{M}$ — повідомлення, що підлягає приховуванню. Тоді модифіковане зображення (стеганоконтейнер) $\mathbf{S} = \mathbf{C} + \mathbf{M}$. Також передбачається, що модифіковане зображення $\mathbf{S}$ є візуально тотожнім до первинного і може бути піддане у стеганоканалі стисненню із втратами: $\mathbf{S}^{\nabla} = \boldsymbol{\Theta}(\mathbf{S})$, де $\boldsymbol{\Theta}$ — оператор стиснення. Завдання одержувача — видобути з отриманого контейнера $\mathbf{S}^{\nabla}$ вбудовані на передавальному боці біти даних $\mathbf{M}_i$ і сформувати копію повідомлення $\mathbf{M}^*$. Блок-схема такого стеганоканалу представлена на рис. 5.27.

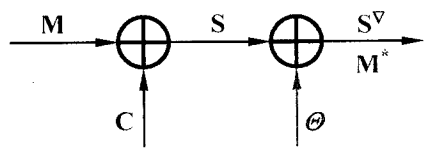


Рис. 5.27. Блок-схема стеганоканалу з атакою стисненням

Основне, що нас при цьому цікавитиме, — це відповідь на питання: яку кількість бітів можна ефективно вбудувати до зображення певного

розміру і згодом видобути з нього за умови задовільно низької ймовірності помилок на останньому етапі. Інакше кажучи, якою є прихована пропускна здатність стеганоканалу при застосуванні у відкритому каналі зв'язку певного алгоритму стиснення?

За окреслених вище умов, повідомлення **M** передається каналом, що має два джерела «шуму»: по-перше, це елементи власне зображення-контейнера **C**, по-друге, це «артефакти» (спотворення цифрових мультимедійних даних), що виникають в результаті операцій стиснення Θ. Отож, на виході системи маємо два цифрових потоки: $\mathbf{S}^{\nabla}$ і $\mathbf{M}^{*}$ — можливо спотвореного стеганоконтейнера і, як наслідок, — повідомлення. Структурна схема такої стеганосистеми наведена на рис. 5.28.

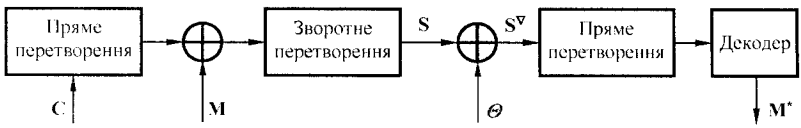


Рис. 5.28. Структурна схема стеганосистеми за дії у каналі атаки стисненням із втратами

На етапі вбудовування даних стеганограми зображення розкладається на J субсмуг (пряме перетворення), до кожної з яких вбудовується певна частина приховуваної інформації (наприклад, один біт). Після зворотного перетворення отримується модифіковане зображення **S**, що потрапляє у відкритий канал зв'язку. У наслідок можливого стиснення Θ, на виході каналу маємо варіант зображення $\mathbf{S}^{\nabla}$, що на приймальному боці знову піддається прямому перетворенню і з кожної з J субсмуг видобуваються складові елементи прихованого повідомлення $\mathbf{M}^{*}$.

Як зазначається у [5], реальні зображення не становлять собою випадкові процеси з рівномірно розподіленими значеннями величин. Відомо, і це використовується в алгоритмах стиснення з втратами, що більша частина енергії зображень зосереджена в низькочастотній (НЧ) частині їхнього спектра. Звідси й виникає потреба у здійсненні декомпозиції зображення на субсмуги, до яких, власне, і вбудовуються біти стеганоповідомлення. НЧ субсмуги містять визначальну частину енергії зображення і, таким чином, носять шумовий (маскуючий) характер. Зі свого боку, високочастотні (ВЧ) субсмуги найбільшим чином піддаються впливові з боку різноманітних алгоритмів обробки, як, наприклад, НЧ-фільтрація, стиснення тощо. Таким чином, для формування стеганограми найоптимальнішими є середньочастотні (СЧ) субсмуги спектра зображення.

Типовий розподіл шуму зображення і шуму обробки по частотах зображено на рис. 5.29 [44, 85].

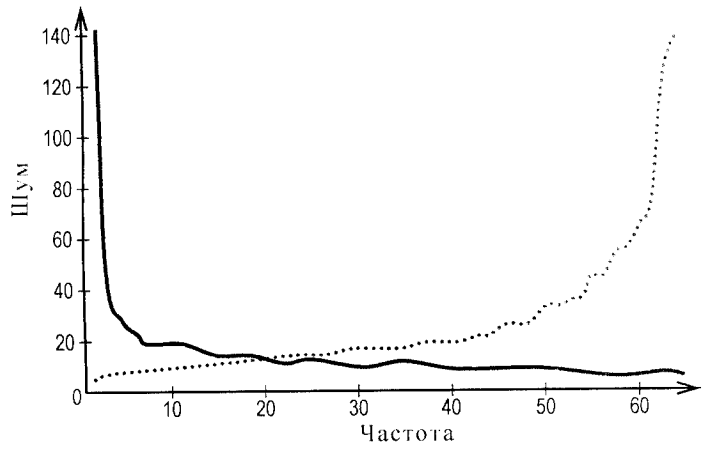


Рис. 5.29. Залежність шуму зображення (суцільна крива) і шуму обробки (пунктирна крива) від частоти

Шляхом виконання прямого і зворотного перетворень стеганоканал можна розкласти на J незалежних субканалів, у кожному з яких існує по два джерела шуму. Позначимо через $\sigma^2_{C_j}$ і $\sigma^2_{\Theta_j}$ дисперсії коефіцієнтів стеганоперетворення (шуму зображення $\mathbf{C}$) і стиснення із втратами (шуму обробки Θ) у j-му субканалі, $j = 1, \ldots, J$. Тоді вираз для прихованої пропускної здатності каналу стеганосистеми набуде вигляду:

$$B = \frac{X \cdot Y}{2 \cdot J} \cdot \sum_{j=1}^{J} log_2 \left(\frac{\nu_j^2}{\sigma^2_{C_j} + \sigma^2_{\Theta_j}} + 1 \right) \text{ біт}, \qquad (5.16)$$

де ν_j — візуальний поріг для j-ї субсмуги (ν_j^2 — максимально припустима енергія стеганоповідомлення, виходячи з вимог збереження візуальної якості зображення); X та Y — піксельний розмір зображення-контейнера $\mathbf{C}$.

Обрання значення візуального порогу ґрунтується на врахуванні властивостей ЗСЛ. Відомо, що шум у ВЧ областях зображення більш прийнятний, аніж в областях НЧ. Отож, відношення «сигнал/шум» (у сенсі: сигнал повідомлення — до шуму зображення) має бути якнайменшим саме для НЧ субсмуг.

У загальному випадку, чим нижчою є частота субсмуги, тим більшою є дисперсія σ^2. Можна ввести певні вагові коефіцієнти:

$$\nu_j^2 = \kappa \cdot \sigma_{*_j}^{2\cdot\alpha},$$

де $*_j$ — узагальнене позначення j-го субканалу **C** або **Θ**; $0 \le \alpha \le 1$ та $\kappa << \sigma_{*_j} \ \forall\ j$ — константи. При цьому випадок $\alpha = 0$ відповідає рівномірному розподілу стеганограми по всім субсмугам, а випадок $\alpha = 1$ — розподілу стеганограми у відповідності до дисперсій останніх.

Для випадку відсутності шуму обробки і якщо припустити, що функції розподілу ймовірностей усіх субканалів є однаковими і такими, що

$$\kappa \cdot \sigma_{*_j} = \kappa_1 \cdot \sigma_{C_j},$$

вираз для пропускної здатності каналу передавання стеганографічно прихованих даних набуває вигляду:

$$B = \frac{X \cdot Y}{2 \cdot J} \cdot \sum_{j=1}^{J} log_2 \left(\frac{\kappa_1 \cdot \sigma_{C_j}^{2\cdot\alpha}}{\sigma_{C_j}^2} + 1 \right) \approx \frac{X \cdot Y}{2 \cdot J} \cdot log_2 \left(1 + \sum_{j=1}^{J} \frac{\kappa_1}{\sigma_{C_j}^{2\cdot(1-\alpha)}} \right). \quad (5.17)$$

Знак наближення у виразі (5.17) є цілком справедливим, оскільки $\kappa_1 \cdot \sigma_{C_j}^{2\cdot\alpha} / \sigma_{C_j}^2 << 1 \ \forall\ j$. Як видно, при $\alpha = 1$ декомпозиція жодним чином не впливатиме на пропускну здатність стеганоканалу. Разом з тим, при $\alpha < 1$ пропускна здатність зростатиме за рахунок того, що до субсмуг з низькою дисперсією (області ВЧ) додається відносно більше енергії сигналу приховуваного повідомлення.

У [85] було проведено численні експерименти, що дозволили авторам сформулювати певні рекомендації стосовно обрання того чи іншого виду перетворення для стеганографії. Відомо, що останні можна впорядкувати за досяжними виграшами від алгоритму кодування (рис. 5.30). При цьому під виграшем від кодування розуміється ступінь перерозподілу дисперсій коефіцієнтів перетворення.

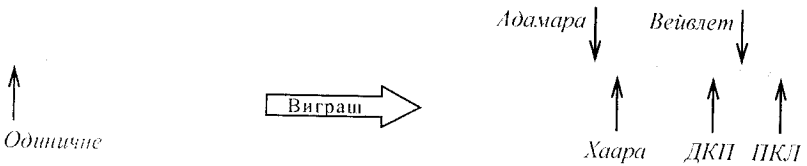


Рис. 5.30. Види перетворень, упорядковані за досяжним виграшем від використовуваного алгоритму кодування

Найбільший виграш дає перетворення Карунена-Лоева (ПКЛ), найменший — розкладання за базисом одиничного імпульсу (тобто відсутність перетворення).

Перетворення, що мають високі значення виграшу від кодування, такі як ДКП, вейвлет-перетворення тощо, характеризуються різко нерівномірним розподілом дисперсій коефіцієнтів субсмуг. ВЧ субсмуги погано підходять для вбудовування через великий шум можливої обробки, а НЧ — через високий шум самого зображення (див. рис. 5.29). Тому, як зазначається у [85], доцільно обмежуватися СЧ смугами, у яких шум зображення вважається порівнянним із шумом обробки. Оскільки загалом таких смуг небагато, то пропускна здатність стеганоканалу, створюваного у частотній області контейнера, виявляється досить малою. У випадку застосування перетворення з більш низьким виграшем від кодування, наприклад, перетворення Адамара або ж Фур'є, існує більше блоків, у яких шум зображення приблизно дорівнює шуму обробки, а, отже, буде вищою і підсумкова пропускна здатність. Висновок, до якого прийшли автори [85], є певною мірою несподіваним: для підвищення прихованої пропускної здатності стеганографічного каналу доцільно застосовувати перетворення з меншими виграшами від кодування, що погано підходять для стиснення сигналів.

У свою чергу, ефективність застосування для стиснення зображень вейвлет-перетворень і ДКП пояснюється тим, що вони добре моделюють процес обробки зображення в ЗСЛ, відокремлюючи значущі деталі від другорядних. Як наслідок, дані перетворення більш доречно застосовувати у випадку наявності активного порушника, оскільки модифікація ним значущих коефіцієнтів приводитиме до неприйнятного спотворення зображення.

У випадку застосування перетворень з низькими виграшами від кодування виникає значна небезпека руйнування вбудованих даних, оскільки коефіцієнти таких перетворень є менш стійкими до модифікацій. Разом з тим, у таких випадках існує більша гнучкість у виборі алгоритму перетворення, і якщо останнє є невідомим порушникові (хоча це й суперечить принципові Керкгоффса), модифікувати стеганограму буде значно складніше.

Під час цифрової обробки зображення найбільш широко застосовується двовимірна версія дискретного косинусного перетворення:

$$\Omega_{\upsilon,\nu} = \frac{\zeta_\upsilon \cdot \zeta_\nu}{\sqrt{2N}} \cdot \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C_{x,y} \cdot \cos\left[\frac{\pi \cdot \upsilon \cdot (2x+1)}{2 \cdot N}\right] \cdot \cos\left[\frac{\pi \cdot \nu \cdot (2y+1)}{2 \cdot N}\right]; \quad (5.18,a)$$

$$S_{x,y} = \frac{1}{\sqrt{2N}} \cdot \sum_{\upsilon=0}^{N-1} \sum_{\nu=0}^{N-1} \zeta_\upsilon \cdot \zeta_\nu \cdot \Omega_{\upsilon,\nu} \cdot \cos\left[\frac{\pi \cdot \upsilon \cdot (2x+1)}{2 \cdot N}\right] \cdot \cos\left[\frac{\pi \cdot \nu \cdot (2y+1)}{2 \cdot N}\right], \quad (5.18,б)$$

де $\mathbf{C}_{x,y}$ і $\mathbf{S}_{x,y}$ — відповідно, елементи первинного і відтвореного за коефіцієнтами ДКП масиву зображення розмірністю $N \times N$; x, y — просторові координати пікселів зображення; $\mathbf{\Omega}_{\upsilon,\nu}$ — коефіцієнт з масиву ДКП; υ, ν — координати коефіцієнта у частотній області; $\zeta_\chi = 1/\sqrt{2}$, якщо $\chi = 0$, і $\zeta_\chi = 1$, якщо $\chi > 0$[XIII].

Розглянемо існуючі методи стеганографічного приховування даних у частотній області зображення, що базуються на алгоритмі ДКП.

5.3.3.1. Метод відносної заміни коефіцієнтів ДКП (метод Коха і Жао)

Один з найпоказовіших методів приховування секретної інформації в частотній області зображення, що його свого часу описали Екгард Кох *(E. Koch)* і Цзян Жао *(J. Zhao)* [86, 87], полягає у відносній зміні величин коефіцієнтів ДКП.

На початковому етапі контейнер-оригінал розбивається на блоки розміром 8×8 пікселів. У результаті застосування до кожного такого блока прямого ДКП (формула (5.18, *a*)) одержуються матриці 8×8 коефіцієнтів розкладу, які зазвичай позначаються $(\mathbf{\Omega}_b)_{\upsilon,\nu}$, де b — номер блока з матриці $\mathbf{C}$, якому відповідає перетворення, а (υ, ν) — позиція коефіцієнта у матриці $\mathbf{\Omega}_b$.

Кожен блок призначено для приховування одного біта даних. При цьому було описано дві реалізації алгоритму, коли псевдовипадково обираються два або ж три коефіцієнти ДКП.

Розглянемо перший варіант. Під час організації секретного каналу абоненти мають попередньо домовитися про конкретні два коефіцієнти ДКП з кожного блока, які використовуватимуться для перенесення даних: (υ_1, ν_1) та (υ_2, ν_2). Певна річ, що задля забезпечення прихованості інформації у досить суттєвих для ЗСЛ областях графічного контейнера і більш-менш достатньої стійкості до можливих атак *JPEG*-стисканням,

---

[XIII] Для випадку, коли індексація елементів масивів починається не з 0, а з 1, формули (5.18) набувають вигляду:

$$\mathbf{\Omega}_{\upsilon,\nu} = \frac{\zeta_\upsilon \cdot \zeta_\nu}{\sqrt{2 \cdot N}} \cdot \sum_{x=1}^{N} \sum_{y=1}^{N} \mathbf{C}_{x,y} \cdot \cos\left[\frac{\pi \cdot (\upsilon - 1) \cdot (2 \cdot x - 1)}{2 \cdot N}\right] \cdot \cos\left[\frac{\pi \cdot (\nu - 1) \cdot (2 \cdot y - 1)}{2 \cdot N}\right];$$

$$\mathbf{S}_{x,y} = \frac{1}{\sqrt{2 \cdot N}} \cdot \sum_{\upsilon=1}^{N} \sum_{\nu=1}^{N} \zeta_\upsilon \cdot \zeta_\nu \cdot \mathbf{\Omega}_{\upsilon,\nu} \cdot \cos\left[\frac{\pi \cdot (\upsilon - 1) \cdot (2x - 1)}{2 \cdot N}\right] \cdot \cos\left[\frac{\pi \cdot (\nu - 1) \cdot (2y - 1)}{2 \cdot N}\right],$$

де $\zeta_\chi = 1/\sqrt{2}$, якщо $\chi = 1$, і $\zeta_\chi = 1$, якщо $\chi > 1$.

зазначені коефіцієнти повинні відповідати косинус-функціям із середніми частотами.

Власне процес приховування починається з псевдовипадкового обрання блока $\mathbf{C}_b$ зображення, призначеного для кодування b-го біта повідомлення. Вбудовування інформації здійснюється таким чином: для передавання «0» прагнуть, щоб різниця абсолютних значень коефіцієнтів ДКП була більшою за деяку додатну величину (поріг), а для передавання «1» ця різниця робиться меншою за від'ємне значення порогу:

$$\begin{cases} \left|\left(\Omega_b\right)_{\nu_1,\nu_1}\right| - \left|\left(\Omega_b\right)_{\nu_2,\nu_2}\right| > P, \text{ при } \mathbf{M}_b = 0; \\ \left|\left(\Omega_b\right)_{\nu_1,\nu_1}\right| - \left|\left(\Omega_b\right)_{\nu_2,\nu_2}\right| < -P, \text{ при } \mathbf{M}_b = 1. \end{cases} \tag{5.19}$$

Таким чином, якщо відносна величина конкретних коефіцієнтів ДКП не відповідає приховуваному бітові, первинне зображення зазнає спотворень у відповідному блоці. Чим більшим було обрано значення порогу P, тим більш стійкою до стиснення буде стеганосистема, але якість зображення при цьому відчутно погіршується.

Після відповідного до (5.19) корегування коефіцієнтів проводиться зворотне ДКП.

Для видобування даних в стеганодекодері виконується та сама процедура обрання обумовлених коефіцієнтів, а рішення про переданий біт приймається згідно наступного правила:

$$\begin{cases} \mathbf{M}_b^* = 0, \text{ при } \left|\left(\Omega_b^*\right)_{\nu_1,\nu_1}\right| > \left|\left(\Omega_b^*\right)_{\nu_2,\nu_2}\right|; \\ \mathbf{M}_b^* = 1, \text{ при } \left|\left(\Omega_b^*\right)_{\nu_1,\nu_1}\right| < \left|\left(\Omega_b^*\right)_{\nu_2,\nu_2}\right|. \end{cases} \tag{5.20}$$

Промоделюємо описаний метод у програмі *Mathcad*.

Крок 1

Виділяємо масиви колірних компонент контейнера-оригіналу:

R := **READ_RED**("C.bmp");
G := **READ_GREEN**("C.bmp");
B := **READ_BLUE**("C.bmp").

У зв'язку з низькою чутливістю ЗСЛ до синього кольору і передбачаючи доволі значне за певних обставин спотворення контейнера при його заповненні, секретне повідомлення вбудовуватимемо до масиву **B**.

Визначимо розмірність масиву **B** і задамо розмірність блоків (сегментів), на які він розбиватиметься: кількість рядків **X** := **rows**(**B**) = 128;

кількість стовпців $\mathbf{Y} := \mathbf{cols}(\mathbf{B}) = 128$; розмірність сегментів $\mathbf{N} := 8$ пікселів. Загальна кількість сегментів, на яку при цьому розбивається зображення: $\mathbf{N_S} := \mathbf{X} \cdot \mathbf{Y} / \mathbf{N}^2 = 256$ сегментів.

Розбиття масиву **B** на сегменти $\mathbf{C_b}$ проводимо за допомогою модуля, аналогічного (М.34), з тією лише відмінністю, що задля узгодженості з виразами (5.18) замість позначення блоків через **Б** ми використовуватимемо позначення **C**.

Крок 2

Оскільки кожен сегмент $\mathbf{C_b}$ призначено для приховування одного біта повідомлення **M**, то попередньо, для можливості виконання цієї операції, необхідно перевірити достатність кількості отриманих сегментів.

Приховуване нами повідомлення, що, як і у попередніх випадках, має вигляд **M** := "© Alex Puzyrenko, 2017", складається з 22 символів. Таким чином, враховуючи, що один символ кодується одним байтом, бітова довжина повідомлення становить $\mathbf{L_M} := 8 \cdot \mathbf{strlen}(\mathbf{M}) = 176$ бітів і умова $\mathbf{L_M} \le \mathbf{N_S}$ виконується.

Крок 3

Застосуємо до кожного з сегментів пряме дискретне косинусне перетворення, скориставшись для цього програмним модулем (М.43) і вбудованою функцією *Mathcad* **dct2d**(...), в основу якої, у свою чергу, покладено реалізацію виразу (5.18, *а*).

$$\Omega := \left| \begin{array}{l} \textbf{for}\ \ b \in 1 .. \mathbf{N_S} \\ \quad \Omega_b \leftarrow \mathbf{dct2d}(\mathbf{C_b}) \\ \Omega \end{array} \right. \qquad \text{(М.43)}$$

Для кожного сегмента $\mathbf{C_b}$ обчислюється матриця Ω_b спектральних коефіцієнтів ДКП. Розмірність такої матриці відповідає розмірності блока, на основі якого вона була визначена, — $\mathbf{N} \times \mathbf{N}$.

Елемент $(\Omega_b)_{1,1}$ у лівому верхньому куті матриці (нагадаємо, що в нашому випадку індексація елементів масивів починається з одиниці) містить інформацію про яскравість всього сегмента і його зазвичай називають постійною *DC*-компонентою. Решта $(\mathbf{N} \cdot \mathbf{N} - 1)$ елементів звуться змінними *AC*-компонентами. Відзначимо також, що коефіцієнти НЧ компонент розташовані ближче до лівого верхнього, а коефіцієнти ВЧ компонент — ближче до правого нижнього кута матриці ДКП (рис. 5.31).

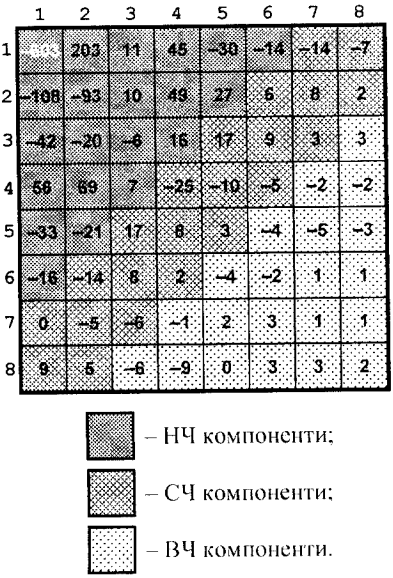


Рис. 5.31. Приклад масиву Ω_b коефіцієнтів ДКП

Крок 4

Як уже зазначалося вище, НЧ компоненти містять переважну частину енергії зображення і, отже, носять шумовий характер. ВЧ компоненти більше піддаються впливові з боку різноманітних алгоритмів обробки (див., наприклад, рис. 5.29). Саме тому для забезпечення достатньо непомітного і, разом з тим, стійкого вбудовування до графічного контейнера сторонніх бітів більш оптимальними є середньочастотні компоненти його спектра.

Задамо позиції двох коефіцієнтів ДКП у масиві Ω_b, що використовуватимуться нами при формуванні стеганошляху. Для забезпечення прихованості стеганограми у достатньо суттєвих для якості контейнера областях, і певної стійкості до *JPEG*-стиснення ці коефіцієнти повинні відповідати косинус-функціям із середніми частотами. Крім того, для створення більш захищеної від зламу стеганосистеми зазначена пара коефіцієнтів з множини СЧ має обиратися псевдовипадково.

Нехай перший коефіцієнт визначається парою координат $\upsilon_1 := 3$ і $\nu_1 := 7$, а другий — $\upsilon_2 := 7$ і $\nu_2 := 3$.

Також встановимо значення порогу, з яким порівнюватимуться різниці модулів коефіцієнтів. Нехай $\mathbf{P} := 23$.

Крок 5

Вбудовування повідомлення проводитимемо у відповідності до рекомендацій, сформульованих перед виразом (5.19): для передавання біта «0» необхідно, щоб різниця модулів коефіцієнтів ДКП була більшою за величину **P**, а для передавання біта «1» — меншою за **–P**. Даний принцип покладено в основу модуля (М.44).

$$\Omega_M := \left|\begin{array}{l} \Omega_M \leftarrow \Omega \\ M \leftarrow \mathrm{str2vec}(M) \\ b \leftarrow 1 \\ \text{for } \mu \in 1..\mathrm{rows}(M) \\ \quad m \leftarrow \mathrm{D2B}(M_\mu, 8) \\ \quad \text{for } i \in 1..8 \\ \quad\quad \Omega' \leftarrow \Omega_{i+N\cdot(\mu-1)} \\ \quad\quad \text{for } n \in 1..2 \\ \quad\quad\quad \omega_n \leftarrow \left|\Omega'_{\upsilon_n,\nu_n}\right| \\ \quad\quad\quad z_n \leftarrow \mathrm{sign}\left(\Omega'_{\upsilon_n,\nu_n}\right) \\ \quad\quad \omega_1 \leftarrow P+\omega_2+1 \quad \text{if } m_i = 0 \wedge \omega_1-\omega_2 \le P \\ \quad\quad \omega_2 \leftarrow P+\omega_1+1 \quad \text{if } m_i = 1 \wedge \omega_1-\omega_2 \ge -P \\ \quad\quad \text{for } n \in 1..2 \\ \quad\quad\quad \Omega'_{\upsilon_n,\nu_n} \leftarrow z_n\cdot\omega_n \\ \quad\quad \Omega_{M_b} \leftarrow \Omega' \\ \quad\quad b \leftarrow b+1 \\ \Omega_M \end{array}\right. \qquad \text{(М.44)}$$

На початку модуля масиву Ω_M присвоюється значення масиву Ω, а рядок символів **M** перетворюється на вектор їхніх *ASCII*-кодів. У циклі зміни μ (перебирання кодів символів) формат коду кожного символу перетворюється з десяткового на двійковий. Кожен з вісімки отриманих при цьому бітів приховується в окремому блоці зображення шляхом модифікації значень коефіцієнтів ДКП відповідного сегмента.

Для зменшення помітності вбудовування, замість односторонньої зміни величини одного з двох коефіцієнтів ДКП для задоволення вимог (5.19), можна змінювати їх одночасно у протилежних напрямках: наприклад, при вбудовуванні «0» — збільшувати модуль коефіцієнта з координатами (υ_1, ν_1) і одночасно зменшувати модуль коефіцієнта з координатами (υ_2, ν_2).

Крок 6

Виконаємо зворотне ДКП (М.45) відповідно до формули (5.18,*б*), використовуючи вбудовану функцію **idct2d**(...).

Оскільки модифікація коефіцієнтів ДКП у деяких випадках може призвести до виходу значень інтенсивностей пікселів зображення за припустимі межі [0; 255], за допомогою вбудованої функції **clip**(...) наприкінці тіла циклу виконується обрізка недопустимих значень.

$$C_M := \left| \begin{array}{l} \text{for } b \in 1..N_S \\ \quad \left| \begin{array}{l} c \leftarrow \text{idct2d}\left(\Omega_{M_b}\right) \\ C_{M_b} \leftarrow \text{clip}(c, 0, 255) \end{array} \right. \\ C_M \end{array} \right. \qquad \text{(M.45)}$$

Крок 7

Проведемо збирання сегментів C_{M_b} до масиву B_M, використовуючи програмний модуль (М.46).

Графічна інтерпретація отриманого в результаті масиву синього кольору, а також підсумковий контейнер-результат, відновлений за колірними складовими **R-G-B**$_M$, представлені на рис. 5.32,*а*. Крім того, на рис. 5.32,*б* наведені приклади масивів синього кольору і контейнера-результату у випадку встановлення на порядок вищого значення порогу **P** за незмінних значень (υ_1, ν_1) і (υ_2, ν_2).

$$B_M := \left| \begin{array}{l} B'_M \leftarrow 0 \\ \text{for } b \in 1..N_S \\ \quad \left| \begin{array}{l} B'_M \leftarrow \text{if}\left(B'_M = 0, C_{M_b}, \text{stack}\left(B'_M, C_{M_b}\right)\right) \\ \text{if } \neg\text{mod}\left(b, \frac{X}{N}\right) \\ \quad \left| \begin{array}{l} B_M \leftarrow \text{if}\left(b = \frac{X}{N}, B'_M, \text{augment}\left(B_M, B'_M\right)\right) \\ B'_M \leftarrow 0 \end{array} \right. \end{array} \right. \\ B_M \end{array} \right. \qquad \text{(M.46)}$$

Крок 8

Розглянемо процес видобування повідомлення на приймальному боці.

Одержувачеві мають бути відомими: алгоритм приховання, колірний масив-контейнер **B*** розмірності **X***×**Y***, розмірність сегментів **N***×**N***, а також матричні координати (υ_1^*, ν_1^*) і (υ_2^*, ν_2^*) коефіцієнтів косинусних функцій, що використовувалися під час приховання.

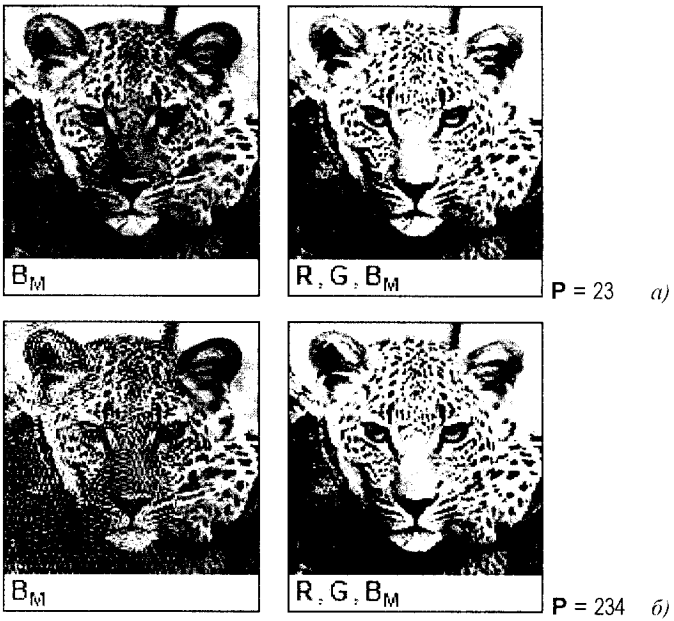


Рис. 5.32. Приклади результатів вбудовування повідомлення **M** до контейнера модифікацією коефіцієнтів ДКП у каналі синього при різних значеннях порогу **P**

На підставі цих даних визначається загальна кількість N^*_S сегментів у контейнері, а масив B^* розбивається на блоки (див. крок № 1). У підсумку одержується масив C^*, кожен елемент якого становить собою матрицю-сегмент розмірністю $N^* \times N^*$.

До кожного елемента масиву C^* застосовується пряме ДКП (див. модуль (М.43)) і для кожного окремого блока контейнера визначається масив Ω^* коефіцієнтів його ДКП.

Видобування прихованої інформації здійснюється у модулі (М.47), до основи якого покладено циклічну перевірку співвідношень (5.20), за результатами якої поточному бітові m^*_i формованої копії повідомлення присвоюється значення 0 або 1. Після видобування кожного восьмого біта вектор m^* двійкових даних перетворюється на відповідне десяткове число — *ASCII*-код, що заноситься до визначеного змінною μ елемента вектора M^* (для коректності обробки у *Mathcad* деяких технічних символів, проводиться їх заміна на символ пробілу з кодом 32). Наприкінці модуля результуючий вектор *ASCII*-кодів трансформується на символьний рядок.

```
M* := | i ← 1,  μ ← 1
      | for  b ∈ 1.. N*_S
      |   Ω' ← Ω*_b
      |   for  n ∈ 1.. 2
      |     ω_n ← |Ω'_{υ*_n, ν*_n}|
      |   m*_i ← if(ω_1 > ω_2, 0, 1)
      |   i ← i + 1  if  rows(m*) < 8
      |   otherwise
      |     i ← 1
      |     ASCII ← B2D(m*)
      |     M*_μ ← if(ASCII < 32, 32, ASCII)
      |     m* ← 0
      |     μ ← μ + 1
      | vec2str(M*)
```

(M.47)

За наведених вище параметрів, наприклад, було отримано наступний текстовий рядок:

$$\mathbf{M}^* = \text{"© Alex Puzyrenko, 2017μ Wà\%d"}$$

Крок 9

Результати обчислення показників візуального спотворення при ($\upsilon_1 = 3$; $\nu_1 = 7$) і ($\upsilon_2 = 7$; $\nu_2 = 3$) для двох значень порогу $\mathbf{P} = 1$ і $\mathbf{P} = 23$ зведені до табл. 5.11 (стор. 234).

Крок 10

Результати видобування повідомлення **M**, вбудованого до НЧ-, СЧ- або ВЧ-областей масивів коефіцієнтів ДКП за однакового значення порогу $\mathbf{P}=23$, зі стиснутого за алгоритмом *JPEG* контейнера зведені до табл. 5.7.

Таблиця 5.7

Залежність *BER* від областей частот вбудовування і рівня збереженої якості при *JPEG* стисненні

Якість, %		100	95	90	85	80	75	65	50	30	0
BER, %	НЧ: (3;1) і (1;3)	0.000	0.000	0.000	0.568	1.136	1.705	2.841	10.795	18.750	25.568
	СЧ: (3;7) і (7;3)	0.000	0.000	0.000	14.733	27.273	32.955	42.614	46.591	44.886	50.568
	ВЧ: (8;6) і (6;8)	0.000	0.000	1.705	21.023	40.909	40.909	50.568	51.705	46.591	50.568

Крок 11

Результати видобування повідомлення **M**, вбудованого до СЧ-областей масивів коефіцієнтів ДКП (3;7) і (7;3) за різних значень порогу **P**, зі стиснутого за алгоритмом *JPEG* контейнера зведені до табл. 5.8.

Таблиця 5.8

Залежність *BER* від порогу вбудовування
і рівня збереженої якості при *JPEG* стисненні

Якість, %		100	95	90	85	80	75	65	50	30	0
BER, %	P = 1	14.205	30.114	39.205	41.477	44.318	45.455	47.159	48.295	49.432	47.159
	P = 10	0.000	7.386	26.136	30.682	42.614	39.773	44.886	46.023	44.886	52.841
	P = 100	0.000	0.000	0.000	0.000	0.000	0.000	0.000	5.114	42.045	44.886

5.3.3.2. Метод Бенгема-Мемона-Ео-Йєнґ

Дейв Бенгем *(D. Benham)*, Насір Мемон *(N. Memon)*, Бун-Лок Ео *(B.-L. Yeo)* та Мінерва Йєнг *(M. Yeung)* [103] запропонували оптимізовану версію попереднього методу. Причому оптимізація проводилася ними за двома напрямками:

- для вбудовування було запропоновано використовувати не всі сегменти (блоки) контейнера, а тільки ті, що найбільш для цього придатні;
- у частотній області окремо взятого блока для перенесення вбудовуваного біта повідомлення обираються вже не два, а три коефіцієнти ДКП, що, як буде показано далі, істотно зменшує спотворення контейнера.

Розглянемо зазначені удосконалення більш докладно.

Придатними для вбудовування приховуваної інформації вважаються ті сегменти зображення, що одночасно задовольняють наступним двом вимогам:

1) у сегменті відсутні різкі перепади яскравості;

2) сегмент не є занадто монотонним.

Сегменти, що не відповідають першій вимозі, характеризуються наявністю декількох занадто великих значень НЧ коефіцієнтів ДКП, порівнянних за своєю величиною з *DC*-компонентою. Для блоків, що не задовольняють другій вимозі, є характерною рівність нулеві більшості ВЧ коефіцієнтів. Таким чином, вказані особливості виступають критерієм відбраковування елементів контейнера, непридатних для заповнення.

Зазначені вимоги відбраковування враховуються використанням двох порогових коефіцієнтів: P_{L} (для першої вимоги) і P_{H} (для другої вимоги), перевищення (P_{L}) або недосягнення (P_{H}) яких вказуватиме на те, що візуальна помітність модифікації сегмента у частотній області буде надзвичайно високою, через що останній для перенесення біта повідомлення є непридатним.

Вбудовування біта повідомлення до блока здійснюється у наведений нижче спосіб. З блока, приналежного СЧ області, обираються (для більшої стійкості стеганосистеми — псевдовипадково) три коефіцієнти ДКП з координатами (υ_1, ν_1), (υ_2, ν_2) та (υ_3, ν_3) відповідно. Якщо необхідно провести вбудовування «0», ці коефіцієнти змінюються таким чином, щоб третій коефіцієнт став менше кожного з перших двох; якщо ж потрібно приховати «1», то коефіцієнт з координатами (υ_3, ν_3) робиться більшим за інші:

$$\left\{\begin{array}{l} \left(\Omega_b\right)_{\upsilon_3,\nu_3} < \left(\Omega_b\right)_{\upsilon_1,\nu_1}; \\ \left(\Omega_b\right)_{\upsilon_3,\nu_3} < \left(\Omega_b\right)_{\upsilon_2,\nu_2}. \end{array}\right\} \text{ при } \mathbf{M}_b = 0; \qquad \left\{\begin{array}{l} \left(\Omega_b\right)_{\upsilon_3,\nu_3} > \left(\Omega_b\right)_{\upsilon_1,\nu_1}; \\ \left(\Omega_b\right)_{\upsilon_3,\nu_3} > \left(\Omega_b\right)_{\upsilon_2,\nu_2}. \end{array}\right\} \text{ при } \mathbf{M}_b = 1. \tag{5.21}$$

Як і у попередньому методі (Коха і Жао), для прийняття рішення про достатність відмінності між собою зазначених коефіцієнтів ДКП, до виразу (5.21) вводиться значення порогу розрізнення P:

$$\left\{\begin{array}{l} \left(\Omega_b\right)_{\upsilon_3,\nu_3} < \min\left[\left(\Omega_b\right)_{\upsilon_1,\nu_1}, \left(\Omega_b\right)_{\upsilon_2,\nu_2}\right] - P, \quad \text{при } \mathbf{M}_b = 0; \\ \left(\Omega_b\right)_{\upsilon_3,\nu_3} > \max\left[\left(\Omega_b\right)_{\upsilon_1,\nu_1}, \left(\Omega_b\right)_{\upsilon_2,\nu_2}\right] + P, \quad \text{при } \mathbf{M}_b = 1. \end{array}\right. \tag{5.22}$$

У випадку, якщо така модифікація призводить до занадто великої деградації зображення, коефіцієнти $\mathbf{\Omega}_b$ залишають без змін, а сам блок в якості контейнера не використовується.

За ствердженням авторів [103], використання трьох коефіцієнтів ДКП замість двох і, що найголовніше, відмова від модифікації у випадку неприйнятних спотворень зображення, суттєво зменшує помітність стеганограми. Одержувач, у свою чергу, завжди може визначити блоки, до яких не проводилося вбудовування, просто повторивши аналіз, аналогічний виконаному на передавальному боці[XIV].

[XIV] Це, водночас, є слабким місцем алгоритму по відношенню до активних атак — навіть незначна зміна контейнера на шляху до одержувача може призвести до появи геть інших наборів придатних і непридатних блоків.

Наведемо один з можливих варіантів реалізації даного методу у програмі *Mathcad*.

Кроки 1...3

Первинні дані, програмні модулі розбиття масиву контейнера на блоки та проведення прямого ДКП є аналогічними використаним при моделюванні попереднього методу.

Крок 4

Задамо координати трьох коефіцієнтів ДКП у масиві Ω_b, які будемо використовувати для вбудовування і видобування бітів повідомлення до/з контейнера. Нехай, наприклад, перший коефіцієнт визначається парою координат $\upsilon_1 := 7$ і $\nu_1 := 3$, другий — $\upsilon_2 := 5$ і $\nu_2 := 5$, а третій — $\upsilon_3 := 3$ і $\nu_3 := 7$.

Встановимо також значення порогів відбраковування блоків: нехай $P_L := 1500$, $P_H := 75$. Власне поріг розрізнення вбудованих бітів залишимо на тому ж рівні, що й при моделюванні попереднього методу: $P := 23$.

Крок 5

Побітове вбудовування повідомлення **M** до сегментів контейнера виконуватимемо, використовуючи програмні модулі (М.48)–(М.50).

Програмні модулі (М.48) і (М.49) слугують для підрахунку сум модулів *AC*-компонент, що відповідають, відповідно, низькочастотній (**LF**) і високочастотній (**HF**) ділянкам матриці коефіцієнтів ДКП (див., наприклад, рис. 5.31). При цьому *DC*-компонента до уваги не береться (умова $u + v > 2$).

$$\begin{aligned}
&\mathrm{LF_counter}(\Omega) := \\
&\quad \aleph \leftarrow \mathrm{rows}(\Omega),\ \Sigma_{LF} \leftarrow 0,\ v\$ \leftarrow \aleph - 1 \\
&\quad \text{for } u \in 1..\aleph - 1 \\
&\qquad \text{for } v \in 1..v\$ \\
&\qquad\quad \Sigma_{LF} \leftarrow \Sigma_{LF} + \mathrm{round}\left(\left|\Omega_{u,v}\right|\right) \text{ if } (u+v) > 2 \\
&\qquad\quad v\$ \leftarrow v\$ - 1 \text{ if } v = v\$ \\
&\quad \Sigma_{LF}
\end{aligned} \tag{М.48}$$

$$\begin{aligned}
&\mathrm{HF_counter}(\Omega) := \\
&\quad \aleph \leftarrow \mathrm{rows}(\Omega),\ \Sigma_{HF} \leftarrow 0,\ v\$ \leftarrow \aleph \\
&\quad \text{for } u \in 3..\aleph \\
&\qquad \text{for } v \in v\$..\aleph \\
&\qquad\quad \Sigma_{HF} \leftarrow \Sigma_{HF} + \mathrm{round}\left(\left|\Omega_{u,v}\right|\right) \\
&\qquad\quad v\$ \leftarrow v\$ - 1 \text{ if } v = v\$ \\
&\quad \Sigma_{HF}
\end{aligned} \tag{М.49}$$

$\Omega_M :=$ | Ⓘ — див. (М.21), крім $S \leftarrow C$ (М.50)

$\Omega_M \leftarrow \Omega,\ w \leftarrow 0$

for $j \in 1..L_M$

$w \leftarrow w+1$

if $w > N_S$

$\Omega_M \leftarrow$ "Замало придатних блоків! Розширте межі!"

break

for $b \in w..N_S$

$\Omega' \leftarrow \Omega_b$

$\Sigma_{LF} \leftarrow LF\_counter(\Omega')$

$\Sigma_{HF} \leftarrow HF\_counter(\Omega')$

$w \leftarrow b,$ break if $\Sigma_{LF} < P_L \wedge \Sigma_{HF} > P_H$

for $n \in 1..3$

$\omega_n \leftarrow \Omega'_{\upsilon_n,\nu_n}$

$\begin{pmatrix} k \\ \omega' \end{pmatrix} \leftarrow if\left[M_{vec\_bin_j}, \begin{pmatrix} -0.5 \\ max(\omega) \end{pmatrix}, \begin{pmatrix} 0.5 \\ min(\omega) \end{pmatrix}\right]$

for $n \in 1..2$

$\Omega'_{\upsilon_n,\nu_n} \leftarrow \omega_n + k \cdot P$

$\Omega'_{\upsilon_3,\nu_3} \leftarrow \omega' - k \cdot P$

$\Omega_{M_w} \leftarrow \Omega'$

Ω_M

замінити 0.5 на 1

видалити

При зміні лише ω_3

В основному модулі вбудовування — (М.50) — після перетворення формату повідомлення з рядка символів **M** на вектор двійкових даних $\mathbf{M}_{vec\_bin}$ розмірністю $L_M \times 1$ відбувається модифікація придатних для заповнення блоків. Перед початком обчислення, масиву Ω_M присвоюються значення первинного немодифікованого масиву Ω коефіцієнтів ДКП для всіх блоків зображення. Перед циклом перебирання елементів вектора $\mathbf{M}_{vec\_bin}$ змінна-лічильник **w** приймає нульове значення. На початку циклу здійснюється перевірка умови невиходу змінної **w** за межі загальної кількості сегментів N_S: якщо $w > N_S$, вбудовування переривається після присвоєння змінній Ω_M рядка символів, що інформує відправника про недостатність придатних для заповнення сегментів (при поточних значеннях порогів P_L і P_H) і рекомендує розширити границі (тобто збільшити значення P_L і/або зменшити P_H), що призведе до зростання кількості блоків, які відповідатимуть поставленим вимогам.

Певна річ, такий крок спричинить появу у списку обраних для заповнення блоків, що будуть для цього не в достатній мірі задовільними. Тому в ході домовленості між сторонами стеганографічного обміну стосовно алгоритму псевдовипадкового вибору трьох пар координат коефіцієнтів ДКП та значень порогів відбраковування обов'язково має бути присутньою фаза перевірки обраного в якості контейнера зображення (або набору зображень) на достатність його пропускної здатності. Іншими словами, після встановлення певних значень порогів $\mathbf{P_L}$ і $\mathbf{P_H}$ обчислюється кількість блоків, визнаних придатними для вбудовування, та виконується оцінка візуального спотворення контейнера. За одержаними результатами приймається рішення або про достатність обраних значень порогів, або ж про необхідність їх розширення.

Результати обчислення кількості придатних блоків $\mathbf{N_{S_{opt.}}}$ в залежності від параметрів $\mathbf{P_L}$ і $\mathbf{P_H}$ наведені на рис. 5.33. Так, наприклад, при встановлені $\mathbf{P_L} = 1500$ і $\mathbf{P_H} = 75$ придатними для вбудовування в обраному нами контейнері визнаються 190 сегментів з 256.

Отож, повертаючись до (М.50), циклом зміни параметра **b** здійснюється вибір матриці коефіцієнтів ДКП, обчислених для **b**-го блока: $\Omega' \leftarrow \Omega_b$. Для матриці Ω' підраховується сума модулів *AC*-компонент, що відповідають її НЧ і ВЧ ділянкам. Якщо значення одержаних сум задовольняють поставленим вимогам, блок, визнається придатним для використання в якості контейнера, його номер присвоюється змінній **w** і поточний цикл зміни **b** переривається.

Для масиву $\Omega' = \Omega_b$, на якому відбулася зупинка, проводиться зчитування коефіцієнтів що відповідають координатам $(\upsilon_1; \nu_1)$, $(\upsilon_2; \nu_2)$ і $(\upsilon_3; \nu_3)$ — ω_1, ω_2 і ω_3 відповідно. Якщо умова (5.22) не виконується, проводиться модифікація усіх трьох коефіцієнтів. При цьому заміна відбувається таким чином, щоб різниця між ω_3 і будь-яким з інших двох коефіцієнтів не була меншою за поріг розрізнення **P**. Можлива й одностороння модифікація лише коефіцієнта ω_3 (необхідні при цьому заміни схематично зображені на рисунку для модуля (М.50)), але за такого випадку порушення структури зображення стають більш суттєвими.

Модифіковані значення коефіцієнтів замінюють відповідні за координатами значення у масиві Ω', який, у свою чергу, присвоюється **w**-му елементові результуючого масиву Ω_M.

Кроки 6, 7

Використовуючи модулі (М.45) та (М.46), проводимо, відповідно, зворотне ДКП і збирання графічних блоків $\mathbf{C_{M_b}}$ до спільного масиву зображення $\mathbf{B_M}$.

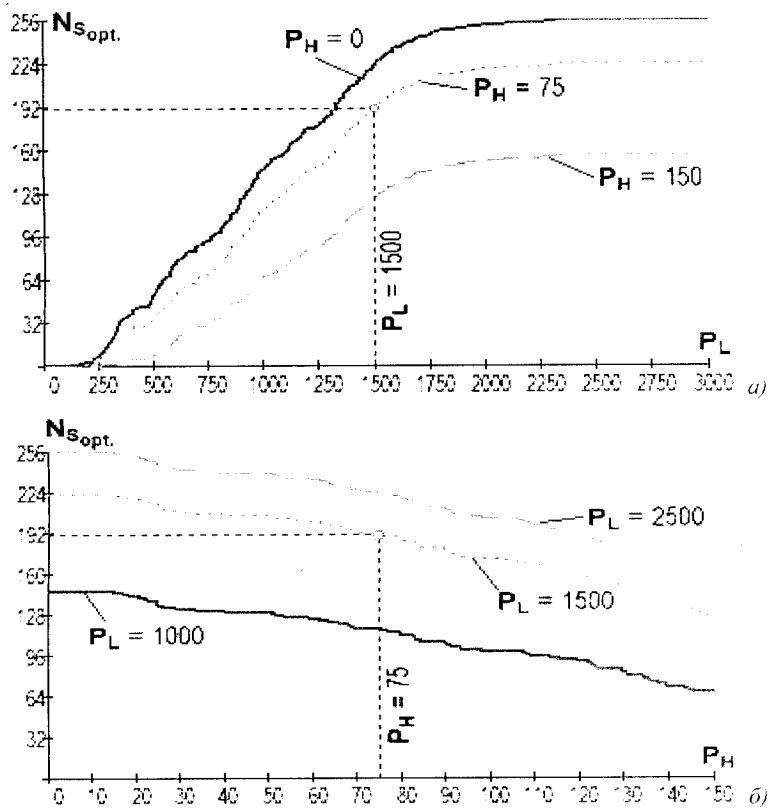


Рис. 5.33. Залежність кількості придатних для вбудовування блоків від значень порогів P_L і P_H: $P_L = var$, $P_H = const$ (*а*); $P_L = const$, $P_H = var$ (*б*)

Графічне представлення відновленого масиву синьої складової при зазначених вище початкових даних наведено на рис. 5.34, *а*. Також на рис. 5.34, *б* представлено вигляд, якого набуває даний масив у випадку встановлення значення порогу розрізнення $P = 234$ за незмінних інших значень (модифікуються усі три коефіцієнти ω), а на рис. 5.34, *в* зображено результат, до якого призводить одностороння модифікація коефіцієнта ДКП ω_3 (див. (М.48)).

Порівнюючи рис. 5.34 з рис. 5.32 за однакових значень порогу P можна зробити висновок про меншу помітність стеганограми, сформованої за методом Бенгема та ін.: сегменти однотонного заповнення, вбудовування до яких було б помітнішим, в якості контейнерів не використовуються.

а) $P = 23$; $\omega_{1(2)}$ і ω_3 *б)* $P = 234$; $\omega_{1(2)}$ і ω_3 *в)* $P = 234$; ω_3

Рис. 5.34. Результати вбудовування повідомлення **M** до масиву каналу синього при обранні оптимальних для вбудовування блоків та різних значеннях і способах забезпечення порогу **P**

Крок 8

На етапі видобування повідомлення повторюється аналіз сегментів, виконаний під час його вбудовування до контейнера.

Попередньо повинні бути відомими: алгоритм приховання, масив-контейнер ($\mathbf{B}^*$) і його розмірність ($\mathbf{X}^*$, $\mathbf{Y}^*$), розмірність сегментів ($\mathbf{N}^*$) та координати коефіцієнтів ДКП, що використовувалися для приховування (або ж алгоритм їх одержання): $(\upsilon_1^*; \nu_1^*)$, $(\upsilon_2^*; \nu_2^*)$ і $(\upsilon_3^*; \nu_3^*)$.

Визначається загальна кількість сегментів у зображенні-контейнері — $\mathbf{N}^*_S$.

Розбивка масиву $\mathbf{B}^*$ на сегменти $\mathbf{C}^*_b$ виконується програмним модулем, ідентичним (М.34).

Крок 9

До кожного сегмента застосовується пряме ДКП (М.43), результатом чого є масив Ω^* коефіцієнтів ДКП сегментів $\mathbf{C}^*_b$.

Крок 10

Видобування прихованої інформації здійснюється програмним модулем (М.51), до основи якого покладено перевірку кожного сегмента зображення на придатність до вбудовування, на підставі чого, у підсумку, вирішується — чи обирався той або інший блок передавальною стороною в якості контейнера для перенесення біта повідомлення. При позитивному рішенні виконується перевірка співвідношень (5.21), за загальними результатами якої формується вектор двійкових даних, що в подальшому може бути перетворений на рядок символів (аналогічно тому, як це здійснювалося у (М.47)).

```
M* := | i ← 1, j ← 1, w ← 0                                          (M.51)
      | for μ ∈ 1.. N*_S
      |   | break if w = N_S
      |   | w ← w + 1
      |   | for b ∈ w.. N*_S
      |   |   | Ω' ← Ω*_b
      |   |   | Σ_LF ← LF_counter(Ω')
      |   |   | Σ_HF ← HF_counter(Ω')
      |   |   | w ← b, break if Σ_LF < P*_L ∧ Σ_HF > P*_H
      |   | for n ∈ 1.. 3
      |   |   ω_n ← Ω'_{υ*_n, ν*_n}
      |   | m*_i ← if[ω_3 ≠ max(ω), 0, if[ω_3 ≠ min(ω), 1, round(rnd(1))]]
      |   | i ← i + 1 if rows(m*) < 8
      |   | if rows(m*) = 8
      |   |   | i ← 1
      |   |   | ASCII ← B2D(m*)
      |   |   | M*_j ← if(ASCII < 32, 32, ASCII)
      |   |   | m* ← 0
      |   |   | j ← j + 1
      | M*
```

За наведених вище параметрів, зокрема, було отримано текстовий рядок **vec2str(M*)** = "© Alex Puzyrenko. 2017 "

Крок 11

Результати обчислення показників візуального спотворення при ($\upsilon_1 = 7$; $\nu_1 = 3$), ($\upsilon_2 = 5$; $\nu_2 = 5$) і ($\upsilon_3 = 3$; $\nu_3 = 7$), $P_L = 1500$ і $P_H = 75$, отримані для двох порогових різних значень $P = 1$ і $P = 23$, зведено до табл. 5.11 (стор. 234).

5.3.3.3. Метод Сю і Ву

Цю-Тінг Сю *(C.-T. Hsu)* і Я-Лінг Ву *(J.-L. Wu)* [104] запропонували алгоритм вбудовування до масивів коефіцієнтів ДКП блоків зображення-контейнера елементів цифрових водяних знаків (ЦВЗ). Наведемо основні положення, закладені авторами до основи свого методу.

Нехай **C** — первинне півтонове зображення розміром $X \times Y$, а **W** — ЦВЗ, що являє собою двійкове (чорно-біле) зображення розміром $A \times Z$. У бінарному ЦВЗ кожен піксель може приймати значення або «1», або

«0», де, наприклад, одиниці відповідає інтенсивність, рівень якої є близьким до 255 (білий піксель), а нулеві — інтенсивність, близька до 0 (чорний піксель).

Оскільки, як буде показано далі, під час вбудовування ЦВЗ оброблятиметься лише СЧ діапазон сигналу-контейнера, необхідною передумовою є те, щоб розмір ЦВЗ був меншим за розмір контейнера. Так, наприклад, для контейнера, розбитого на блоки 8×8, для вбудовування ЦВЗ оптимальним є використання коефіцієнтів ДКП у кількості $64 \cdot A \cdot Z / (X \cdot Y)$. У даному випадку відношення $A \cdot Z / (X \cdot Y)$ визначає ту частку інформації $A \cdot Z$, що може бути вбудована до обраного в якості контейнера сегмента зображення (у наведеному випадку — до 64-х коефіцієнтів блока 8×8). Для більшої стійкості та прихованості результатів застосування стеганометоду кількість вбудовуваної інформації намагаються зменшити.

Зображення-контейнер **C** і ЦВЗ **W** представимо як

$$\mathbf{C} = \{c(x, y);\ 1 \le x \le X;\ 1 \le y \le Y\}, \tag{5.23}$$

$$\mathbf{W} = \{w(a, z);\ 1 \le a \le A;\ 1 \le z \le Z\}, \tag{5.24}$$

де $c(x, y) \in \{0, \ldots, 2^L - 1\}$ — інтенсивність пікселя (x, y) зображення-контейнера; L — кількість бітів, що використовується для квантування значень інтенсивностей пікселів зображення-контейнера; $w(a, z) \in \{0, 1\}$ — двійкові значення пікселя (a, z) зображення ЦВЗ.

Контейнер **C** можна розбити на $\frac{X}{8} \times \frac{Y}{8}$ блоків розмірністю 8×8. Для одержання цієї ж кількості блоків, ЦВЗ має бути розбитим на блоки розмірністю $\frac{8 \cdot A}{X} \times \frac{8 \cdot Z}{Y}$. Наприклад, якщо $A = X/2$, а $Z = Y/2$, то розмірність блока ЦВЗ складе 4×4; якщо $A = X/4$, а $Z = Y/4$, — 2×2 і т. д. Для доповнення контейнера і/або ЦВЗ до необхідної розмірності можуть додаватися додаткові стовпці або рядки.

Псевдовипадкова перестановка пікселів ЦВЗ

У першому наближенні кожен блок ЦВЗ вбудовується до СЧ коефіцієнтів ДКП окремого блока контейнера. Тому, замість усього контейнера, кожен блок ЦВЗ буде розсіяний лише по відповідному йому блоці. Очевидно, що за відсутності належного регулювання просторових зв'язків ЦВЗ, атака звичайним масштабуванням заповненого контейнера може з легкістю зруйнувати ЦВЗ.

Для забезпечення стійкості до масштабування, з метою зміни порядку ЦВЗ для розосередження його просторових зв'язків, авторами [104] було запропоновано використати швидкий метод генерації двовимірного ПВЧ:

$$\mathbf{W}_{\text{rnd}} = \textit{permute}(\mathbf{W}); \tag{5.25}$$

$$\mathbf{W}_{\text{rnd}} = \{w_{rnd}(a, z) = w_{rnd}(a', z');\ 1 \le a \le A;\ 1 \le z \le Z\},$$

де піксель (a', z') являє собою переставлений згідно псевдовипадковій перестановці (операція ***permute***) піксель (a, z).

Перестановка блоків ЦВЗ

У відповідності до збільшення рівня прихованості також мають враховуватися і характеристики самого контейнера (відомо, наприклад, що модифікація ВЧ компонент або ділянок з більшою яскравістю буде менш помітною). Подібні, залежні від контейнера властивості можуть використовуватися для перестановки складових елементів вже псевдовипадково змішаного ЦВЗ для одержання більшої відповідності чутливості ЗСЛ.

У [104] блоки контейнера пропонується впорядкувати у відповідності до зміни дисперсій інтенсивностей пікселів (наприклад, по їх зменшенню). У свою чергу, блоки ЦВЗ сортуються за кількістю інформації (тобто за кількістю значущих (одиничних) пікселів). Вид сортування блоків ЦВЗ (за зростанням або за убуванням) має відповідати аналогічній операції над блоками контейнера. У кінцевому підсумку кожному блоку контейнера відповідатиме свій блок ЦВЗ:

$$\mathbf{W}_{\text{sort}} = \textit{permute}(\mathbf{W}_{\text{rnd}}). \tag{5.26}$$

Приклад сортування і перестановки блоків наведено на рис. 5.35.

Індекс блоку контейнера	*Значення дисперсії*
7	67.1
2	61.2
3	53.0
6	41.9
1	32.3
5	14.5
4	7.4

Індекс блоку ЦВЗ	*Кількість одиниць*
4	15
1	13
3	10
2	9
6	8
7	6
5	3

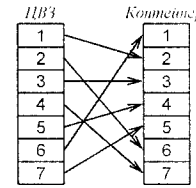


Рис. 5.35. Приклад перестановки блоків ЦВЗ
в залежності від характеристик блоків контейнера

Перетворення блоків контейнера

Оскільки ДКП, використовуване при стисненні *JPEG*, оперує з блоками 8×8, бачиться за доцільне і контейнер **C** розбити на сегменти такої самої розмірності. До кожного блока застосовується пряме ДКП:

$$\mathbf{\Omega} = \textit{FDCT}(\mathbf{C}). \tag{5.27}$$

Обрання середньочастотних коефіцієнтів ДКП

Для того щоб вбудований ЦВЗ був візуально непомітним і, разом з тим, залишався стійким до стиснення даних із втратами, очевидним компромісом буде його вбудовування до діапазону середніх частот контейнера. При цьому для кожного блока 8×8 контейнера з наявних 64-х коефіцієнтів відбираються $64 \cdot A \cdot Z/(X \cdot Y)$ СЧ коефіцієнти ДКП, розміщені вздовж другої діагоналі матриці ДКП.

Для зручності подальших дій відібрані коефіцієнти згортаються до зменшеної матриці розмірністю $\frac{8 \cdot A}{X} \times \frac{8 \cdot Z}{Y}$:

$$\Omega_{\text{mid}} = \textit{reduce}(\Omega). \tag{5.28}$$

Зокрема, якщо $A = X/2$, а $Z = Y/2$, то під час вбудовування ЦВЗ обробляються лише 16 коефіцієнтів ДКП, а інші 48 залишаються без змін. Вищевикладений процес формування масиву СЧ-коефіцієнтів ДКП проілюстровано на рис. 5.36.

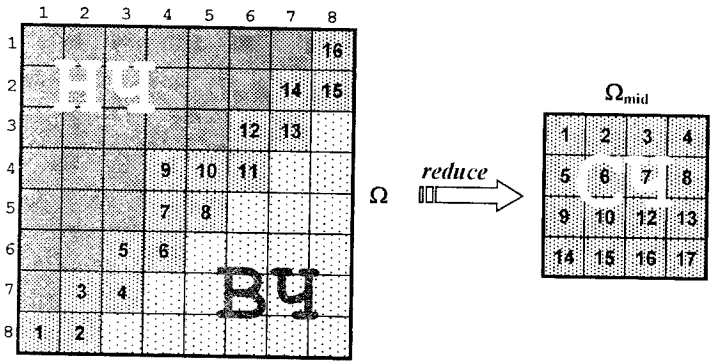


Рис. 5.36. Конфігурація матриці Ω коефіцієнтів ДКП і приклад зведення її СЧ-коефіцієнтів до окремої матриці Ω_{mid}

Модифікація коефіцієнтів ДКП

Результатами виконання попередніх дій є: переставлений у псевдовипадковому порядку і поставлений у відповідність до блока контейнера блок ЦВЗ, а також призведене частотне відображення контейнера (що містить лише СЧ-компоненти первинного зображення), обидва розмірністю $\frac{8 \cdot A}{X} \times \frac{8 \cdot Z}{Y}$.

Елементи масиву $\mathbf{\Omega}_{\text{mid}}$ можуть бути модифіковані у відповідності до пікселів вбудовуваного блока ЦВЗ.

На думку авторів методу, ефективним засобом досягнення непомітності ЦВЗ і стійкості стеганосистеми до атак при низьких коефіцієнтах *JPEG*-стиснення є вбудовування кожного пікселя ЦВЗ шляхом модифікації полярності між відповідними пікселями сусідніх блоків. Проте, як зазначається, такий метод не стійкий до активних атак з високим (≥ 6) коефіцієнтом стиснення.

Пропонується розглянути технічні аспекти проблеми вбудовування при зазначеному підході, а також покращений метод, який є стійкішим до атак стисненням із втратами.

Вбудовування ЦВЗ модифікацією відношень між значеннями коефіцієнтів сусідніх блоків

Для підрахунку полярності обраних СЧ-коефіцієнтів сусідніх блоків використовується так звана "залишкова" маска. На рис. 5.37 зображено приклад такої маски, де кожен елемент (від «А» до «И») містить у собі призведене відображення коефіцієнтів ДКП контейнера $\mathbf{\Omega}_{\text{mid}}$ певного блока, причому позиції «Д» відповідає поточне відображення ДКП.

А	Б	В
Г	**Д**	Е
Ж	З	И

Рис. 5.37. Конфігурація залишкової маски

Якщо, наприклад, А = Б = В = Е = Ж = З = И = 0, Г = -1, а Д = 1, то полярність становитиме собою двійковий образ **P** — масив нулів та одиниць, який вказуватиме на той факт, що коефіцієнт ДКП поточної позиції даного блока відображення коефіцієнтів ДКП є більшим (полярність дорівнює 1) або меншим (полярність дорівнює 0) за коефіцієнт на відповідній позиції попереднього блока. Тобто, для наведеного щойно прикладу:

$$\mathbf{P} = \textit{polarity}(\mathbf{\Omega}_{\text{mid}}) = \begin{cases} 1, \text{ при } \mathbf{\Omega}_{\text{mid}_{b_{u,v}}} > \mathbf{\Omega}_{\text{mid}_{b-1_{u,v}}}; \\ 0, \text{ при } \mathbf{\Omega}_{\text{mid}_{b_{u,v}}} \leq \mathbf{\Omega}_{\text{mid}_{b-1_{u,v}}}. \end{cases} \tag{5.29}$$

де $\mathbf{\Omega}_{\text{mid}_{b_{u,v}}}$ — середньочастотний коефіцієнт ДКП блока $\boldsymbol{b}$.

За інших значень елементів залишкової маски відповідно змінюватиметься і вираз (5.29). При обчисленні полярності, порівняння значення коефіцієнта ДКП поточного блока із значеннями відповідних коефіцієнтів декількох сусідніх блоків у більшості випадків дозволяє, крім підвищення рівня захищеності від зламу стеганосистеми, отримати й менше спотворення контейнера.

Після одержання відображень полярності $\mathbf{P}$ для всіх блоків контейнера, виконується пошук коефіцієнтів ДКП, що потребують модифікації для приховання окремого пікселя зі складу псевдовипадково переставленого ЦВЗ. Пошук проводиться у відповідності до залишкової маски, шляхом зміни поточної полярності (оператор ***XOR*** або знак «⊕» — додавання за модулем 2):

$$\hat{\mathbf{P}} = \mathit{XOR}\left(\mathbf{P}, \mathbf{W}_{\text{sort}}\right); \tag{5.30}$$

$$\hat{\mathbf{P}} = \left\{\hat{\mathbf{P}}_{\upsilon,\nu};\ 1 \le \upsilon \le \frac{8 \cdot A}{X};\ 1 \le \nu \le \frac{8 \cdot Z}{Y}\right\},$$

де $\hat{\mathbf{P}}_{\upsilon,\nu} = \begin{cases} 1 - \mathbf{P}_{\upsilon,\nu}, & \text{при } \mathbf{W}_{\text{sort}_{\upsilon,\nu}} = 1; \\ \mathbf{P}_{\upsilon,\nu}, & \text{при } \mathbf{W}_{\text{sort}_{\upsilon,\nu}} = 0. \end{cases} = \mathbf{P}_{\upsilon,\nu} \oplus \mathbf{W}_{\text{sort}_{\upsilon,\nu}}.$

Далі, на основі масивів полярності $\hat{\mathbf{P}}$ для кожного блока контейнера формують масив $\hat{\mathbf{\Omega}}_{\text{mid}}$ модифікованих СЧ-коефіцієнтів ДКП — за умови, щоб різниця між $\mathbf{\Omega}_{\text{mid}}$ та $\hat{\mathbf{\Omega}}_{\text{mid}}$ була зведена до мінімуму або стала меншою за встановлений поріг η (оператор ***expand***):

$$\hat{\mathbf{\Omega}}_{\text{mid}} = \mathit{expand}\left(\hat{\mathbf{P}}\right), \text{ за умови } \sum_{\upsilon,\nu}\left[\mathbf{\Omega}_{\text{mid}_{\upsilon,\nu}} - \hat{\mathbf{\Omega}}_{\text{mid}_{\upsilon,\nu}}\right]^2 < \eta. \tag{5.31}$$

Наприклад, задаючись початковим коефіцієнтом $\hat{\mathbf{\Omega}}_{\text{mid}_{\upsilon_1,\nu_1}} = \mathbf{\Omega}_{\text{mid}_{\upsilon_1,\nu_1}}$, необхідно додавати/віднімати коефіцієнти сусідніх блоків (у відповідності до залишкової маски) таким чином, щоб, провівши згодом операцію, аналогічну (5.29), можна було одержати належну полярність $\hat{\mathbf{P}}_{\upsilon_1,\nu_1}$. Далі слід перейти до наступних коефіцієнтів, почергово змінюючи лише ті з них, що не впливатимуть на полярність попередньо опрацьованих коефіцієнтів.

Для того щоб зменшити деградацію зображення (як неминучий наслідок вбудовування ЦВЗ), автори методу пропонують обчислювати полярність для абсолютних значень коефіцієнтів ДКП, що дозволить гарантовано зберегти знак («+» або «–») модифікованого коефіцієнта.

Крім того, для підвищення стійкості стеганосистеми до *JPEG*-стиснення із втратами, має бути врахований ефект квантування, що використовується у технології *JPEG*. На рис. 5.38,*а* наведено таблицю квантування яскравості, пропоновану стандартом *JPEG*, яка, зазвичай, викликає помітні спотворення (так звані «артефакти») у стиснутому зображенні. На рис. 5.38,*б* зображено дещо іншу таблицю квантування, використовувану у більшості програм, що працюють з *JPEG*. Видно, що значення при цьому є майже вдвічі меншими за відповідні ним у попередній таблиці.

16	11	10	16	24	40	51	**61**
12	12	14	19	26	58	**60**	**55**
14	13	16	24	40	**57**	**69**	56
14	17	22	**29**	**51**	**87**	80	62
18	22	37	**56**	**68**	109	103	77
24	35	**55**	**64**	81	104	113	92
49	**64**	**78**	87	103	121	120	101
72	**92**	95	98	112	100	103	99

а)

8	6	5	8	12	20	26	**31**
6	6	7	10	13	29	**30**	**28**
7	7	8	12	20	**29**	**35**	28
7	9	11	**15**	**26**	**44**	40	31
9	11	19	**28**	**34**	55	52	39
12	18	**28**	**32**	41	52	57	46
25	**32**	**39**	44	52	61	60	51
36	**47**	48	49	56	50	52	50

б)

Рис. 5.38. Приклади таблиць квантування яскравості: стандартна *JPEG* (*а*) та *Image Alchemy*, *Handmade Software Inc.* (*б*)

Заснована на таблиці квантування полярність являє собою результат обчислення різниці між квантованими і згодом деквантованими коефіцієнтами ДКП відповідних блоків. Для тривіального випадку, коли порівняння ведеться з коефіцієнтами попереднього блока (див. числовий приклад до рис. 5.37), формула (5.29) набуває вигляду

$$\mathbf{P}_b = \begin{cases} 1, \text{ при } \left\lfloor \dfrac{\left\lfloor \boldsymbol{\Omega}_{\mathbf{mid}_{b\,\upsilon,\nu}} \right\rfloor}{\mathbf{Q}_{\mathbf{mid}_{\upsilon,\nu}}} \right\rfloor \cdot \mathbf{Q}_{\mathbf{mid}_{\upsilon,\nu}} > \left\lfloor \dfrac{\left\lfloor \boldsymbol{\Omega}_{\mathbf{mid}_{b-1\,\upsilon,\nu}} \right\rfloor}{\mathbf{Q}_{\mathbf{mid}_{\upsilon,\nu}}} \right\rfloor \cdot \mathbf{Q}_{\mathbf{mid}_{\upsilon,\nu}}; \\ 0, \text{ при } \left\lfloor \dfrac{\left\lfloor \boldsymbol{\Omega}_{\mathbf{mid}_{b\,\upsilon,\nu}} \right\rfloor}{\mathbf{Q}_{\mathbf{mid}_{\upsilon,\nu}}} \right\rfloor \cdot \mathbf{Q}_{\mathbf{mid}_{\upsilon,\nu}} \le \left\lfloor \dfrac{\left\lfloor \boldsymbol{\Omega}_{\mathbf{mid}_{b-1\,\upsilon,\nu}} \right\rfloor}{\mathbf{Q}_{\mathbf{mid}_{\upsilon,\nu}}} \right\rfloor \cdot \mathbf{Q}_{\mathbf{mid}_{\upsilon,\nu}}, \end{cases} \quad (5.32)$$

де $\mathbf{Q}_{\mathbf{mid}_{\upsilon,\nu}}$ — значення квантування для СЧ-коефіцієнта з координатами (υ,ν); дужки $\lfloor ... \rfloor$ вказують на те, що повертається ціла частина від результату ділення.

При цьому, у випадку атаки квантуванням, попереднє урахування ефекту останнього значно підвищує ймовірність правильного розпізнання ознак пікселів під час видобування ЦВЗ. Проте, оскільки квантування має тенденцію до зведення значень багатьох коефіцієнтів у нуль (що є особливо характерним для високочастотних коефіцієнтів), деяка частина СЧ-коефіцієнтів ДКП також в підсумку дорівнюватиме нулеві. Крім того, для збереження встановленої полярності навіть після проведення квантування, тим самим значенням пікселя ЦВЗ мають бути модифіковані не лише визначені СЧ-коефіцієнти у поточному блоці, але також і в усіх сусідніх блоках у відповідності до маски залишковості.

Вбудовування ЦВЗ модифікацією відношень між значеннями коефіцієнтів у межах блока

Для подолання описаних вище технічних недоліків, Сю і Ву запропонували замість порівняння з СЧ-коефіцієнтами ДКП сусідніх блоків використовувати постійну *(DC)* компоненту поточного блока. У цьому випадку вираз (5.32) набуває вигляду

$$\mathbf{P}_b = \begin{cases} 1, \text{ при } \left\lfloor \dfrac{\left| \mathbf{\Omega}_{\mathbf{mid}_{b_{\upsilon,\nu}}} \right|}{\mathbf{Q}_{\mathbf{mid}_{\upsilon,\nu}}} \right\rfloor \cdot \mathbf{Q}_{\mathbf{mid}_{\upsilon,\nu}} > \left\lfloor \dfrac{\mathbf{\Omega}_{b_{1,1}}}{\psi \cdot \mathbf{Q}_{1,1}} \right\rfloor \cdot \mathbf{Q}_{1,1}; \\ 0, \text{ при } \left\lfloor \dfrac{\left| \mathbf{\Omega}_{\mathbf{mid}_{b_{\upsilon,\nu}}} \right|}{\mathbf{Q}_{\mathbf{mid}_{\upsilon,\nu}}} \right\rfloor \cdot \mathbf{Q}_{\mathbf{mid}_{\upsilon,\nu}} \leq \left\lfloor \dfrac{\mathbf{\Omega}_{b_{1,1}}}{\psi \cdot \mathbf{Q}_{1,1}} \right\rfloor \cdot \mathbf{Q}_{1,1}, \end{cases} \tag{5.33}$$

де ψ— масштабний коефіцієнт; $\mathbf{Q}_{1,1}$ — значення квантування для *DC*.

Зворотне перетворення блоків контейнера

За допомогою оператора ***put*** модифіковані матриці СЧ-коефіцієнтів ($\hat{\mathbf{\Omega}}_{\mathbf{mid}}$) відображуються до загальних матриць коефіцієнтів ДКП ($\mathbf{\Omega}$):

$$\hat{\mathbf{\Omega}} = \boldsymbol{put}(\hat{\mathbf{\Omega}}_{\mathbf{mid}}). \tag{5.34}$$

До результату проведеного об'єднання застосовується зворотне ДКП:

$$\mathbf{S} = \boldsymbol{IDCT}(\hat{\mathbf{\Omega}}). \tag{5.35}$$

Видобування ЦВЗ з контейнера

Окрім зображення з вбудованим ЦВЗ, процес видобування вимагає наявності в одержувача ще й оригінальних зображень контейнера і ЦВЗ.

Обидва зображення (оригінальне — **C**, і досліджуване на наявність вбудованого ЦВЗ — $\mathbf{C}^*$) піддаються прямому ДКП:

$$\boldsymbol{\Omega} = \boldsymbol{FDCT}(\mathbf{C}), \quad \boldsymbol{\Omega}^* = \boldsymbol{FDCT}(\mathbf{C}^*).$$

З отриманих при цьому масивів коефіцієнтів ДКП виокремлюються матриці СЧ-коефіцієнтів, які, у свою чергу, використовуються для одержання відповідних шаблонів полярності:

$$\boldsymbol{\Omega}_{\mathbf{mid}} = \boldsymbol{reduce}(\boldsymbol{\Omega}), \quad \boldsymbol{\Omega}^*_{\mathbf{mid}} = \boldsymbol{reduce}(\boldsymbol{\Omega}^*);$$

$$\mathbf{P} = \boldsymbol{polarity}(\boldsymbol{\Omega}_{\mathbf{mid}}), \quad \mathbf{P}^* = \boldsymbol{polarity}(\boldsymbol{\Omega}^*_{\mathbf{mid}}).$$

Застосовуючи до масивів полярностей **P** і $\mathbf{P}^*$ операцію додавання за модулем 2, одержують двійкові (поки що переставлені у просторі і псевдовипадково змішані) дані:

$$\mathbf{W}^*_{\mathbf{sort}} = \boldsymbol{XOR}(\mathbf{P}, \mathbf{P}^*), \tag{5.36}$$

де $\mathbf{W}^*_{\mathbf{sort}_{u,v}} = \mathbf{P}_{u,v} \oplus \mathbf{P}^*_{u,v}$.

Виконується зворотна просторова перестановка блоків отриманих даних. при цьому індекси відповідних пар блоків контейнера і даних, що видобуваються, можуть бути отримані або шляхом їх зчитування з попередньо збережених у файлі на етапі вбудовування ЦВЗ, або ж безпосередньо при видобуванні, шляхом аналогічних дій над зображенням-оригіналом і зображенням-ЦВЗ:

$$\mathbf{W}^*_{\mathbf{rnd}} = \boldsymbol{arrange}(\mathbf{W}^*_{\mathbf{sort}}).$$

Аналогічно проводиться і зворотна псевдовипадкова перестановка даних в одержаному масиві:

$$\mathbf{W}^* = \boldsymbol{arrange}(\mathbf{W}^*_{\mathbf{rnd}}).$$

У розглянутому алгоритмі дій можна виділити три особливості, що можуть бути використані в якості секретного ключа:

1) початкове число генератора ПВЧ, яке визначатиме перший елемент псевдовипадкової перестановки (довільне ціле число на проміжку $[1, A \cdot Z - 1]$);

2) обрання СЧ-коефіцієнтів ДКП (оскільки треба обрати $64 \cdot A \cdot Z / (X \cdot Y)$ коефіцієнти з 64-х для кожного блока, то для кожного блока можна закріпити свій набір коефіцієнтів);

3) зведення обраних коефіцієнтів до окремої матриці (на рис. 5.36 показано лише один з можливих способів такого відображення).

Розглянемо приклад реалізації методу Сю і Ву у середовищі *Mathcad*.

Крок 1

Нехай зображення-контейнер і зображення-ЦВЗ являють собою графічні файли *C.bmp* та *W.bmp* відповідно (рис. 5.39):

C := **READBMP**("C.bmp");

W := **READBMP**("W.bmp").

Геометричні розміри зазначених зображень при цьому становлять:

X := **rows**(**C**) = 128 пікс., **Y** := **cols**(**C**) = 128 пікс.;

A := **rows**(**W**) = 64 пікс., **Z** := **cols**(**W**) = 64 пікс.

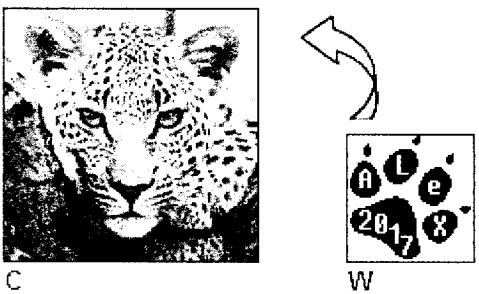


Рис. 5.39. Контейнер-оригінал і вбудовуваний до нього ЦВЗ

Крок 2

Проводимо бінаризацію масиву ЦВЗ (на той випадок, якщо останній є півтоновим зображенням):

W := **binarize**(**W**, 128).

У підсумку, усі елементи вихідного масиву **W** приймають значення 0 або 1.

Крок 3

Розмірність блоків, на які розбивається зображення-контейнер, приймаємо рівною **N** := 8 пікс. Кількість одержуваних при цьому блоків: $\mathbf{N_S} := \mathbf{X} \cdot \mathbf{Y} / \mathbf{N}^2 = 256$.

Розмірність, що її повинні мати блоки ЦВЗ для одержання їх кількості $\mathbf{n_S} = \mathbf{N_S}$, $\mathbf{n} := \mathbf{A} \cdot \mathbf{N} / \mathbf{X} = 4$ пікс.

Виконуємо перевірку кількості одержуваних за таких умов блоків ЦВЗ: $\mathbf{n_S} := \mathbf{A} \cdot \mathbf{Z} / \mathbf{n}^2 = 256$.

Крок 4

Розбиття контейнера **C** на $\mathbf{N_S}$ блоків розмірністю **N**×**N** виконуємо за допомогою програмного модуля, аналогічного (М.34), з тією лише відмінністю, що замість позначення блоків через **Б** будемо використовувати позначення $\mathbf{Б_C}$.

Крок 5

Дотримуючись послідовності дій, запропонованої Сю і Ву, проведемо псевдовипадкову перестановку елементів ЦВЗ. По-перше, проіндексуємо пікселі ЦВЗ (від 1 до $\mathbf{A}\cdot\mathbf{Z} = 4096$), для чого просто розгорнемо масив **W** у вектор (програмний модуль (М.52)).

$$\mathbf{W_{vec}} := \left| \begin{array}{l} \textbf{for}\ \ z \in 1..Z \\ \quad \mathbf{W_{vec}} \leftarrow \textbf{if}\left(z = 1, \mathbf{W}^{\langle 1 \rangle}, \textbf{stack}\left(\mathbf{W_{vec}}, \mathbf{W}^{\langle z \rangle}\right)\right) \\ \mathbf{W_{vec}} \end{array} \right. \qquad \text{(М.52)}$$

По-друге, отримані індекси розставимо у довільному (псевдовипадковому) порядку, для чого використаємо лінійний регістр зсуву зі зворотним зв'язком (ЛРЗЗЗ або *LFSR — Linear Feedback Shift Register*).

Як відомо, ЛРЗЗЗ складається з двох частин: власне регістра зсуву і функції зворотного зв'язку (рис. 5.40) [65]. Регістр зсуву становить собою послідовність бітів (розрядів) $\boldsymbol{R}$, кількість яких $\boldsymbol{d}$ визначається довжиною регістра зсуву. Зворотний зв'язок — сума за модулем 2 визначених бітів регістра (ці біти ще називаються відвідною послідовністю).

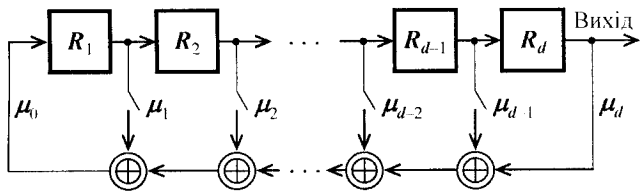


Рис. 5.40. Узагальнений лінійний регістр зсуву із зворотним зв'язком

Теоретично, $\boldsymbol{d}$-бітовий ЛРЗЗЗ може перебувати в одному з 2^d–1 внутрішніх станів, тобто здатен генерувати ПВП з періодом $\boldsymbol{T} = 2^d$–1 бітів. Усі $\boldsymbol{T}$ внутрішніх стани регістр пройде лише за певних відвідних послідовностей. Такі ЛРЗЗЗ мають максимальний період ($\boldsymbol{T} = \boldsymbol{T_M}$), а одержаний при цьому результат називають ***M-послідовністю***.

На рис. 5.40 значення μ_i ($i = 0, 1, \ldots, d$) є ваговим коефіцієнтами поліному $\rho(x)$ степеню d, асоційованого з відвідною послідовністю:

$$\rho(x) = \mu_0 \cdot x^0 + \mu_1 \cdot x^1 + \mu_2 \cdot x^2 + \ldots + \mu_{d-2} \cdot x^{d-2} + \mu_{d-1} \cdot x^{d-1} + \mu_d \cdot x^d.$$

Якщо $\mu_i = 1$, то відповідний ключ замкнений. У випадку $\mu_i = 0$ ключ розімкнений. Невдале ввімкнення суматорів до зворотного зв'язку може призвести до одержання ПВП, період повторення якої буде меншим за максимально можливий при наявній розрядності регістра. Для того, щоб конкретний ЛРЗЗЗ мав максимальний період T_M, поліном $\rho(x)$ має бути примітивним за модулем 2 (тобто не розкладатися на добуток двійкових поліномів менших степенів). При цьому коефіцієнти μ_0 і μ_d повинні завжди дорівнювати 1, оскільки, у випадку $\mu_0 = 0$, поліном $\rho(x)$ ділитиметься на x і не буде примітивним; у випадку ж $\mu_d = 0$, навіть якщо поліном і виявиться примітивним, його степінь буде меншим за d. Решта коефіцієнтів обраного поліному і визначатимуть собою схему формувача ПВП.

У нашому випадку, для перестановки чисел у діапазоні від 1 до **A·Z** є необхідною і достатньою кількість розрядів регістра, що дорівнює **d** := **log**(**A·Z**, 2) = 12. При цьому період повторення ПВП складає **A·Z** – 1 = $= 2^{\mathbf{d}}-1 = 4095$.

Для **d**-розрядного ЛРЗЗЗ в якості примітивного за модулем 2 поліному оберемо наступний: $\rho(x) = 1 + x + x^2 + x^8 + x^{12}$. Цей та деякі інші можливі види примітивних поліномів степеню **d** зведено до табл. 5.9[XV].

Таблиця 5.9

Приклади примітивних за модулем 2 поліномів степеню d = 12

x^0	x^1	x^2	x^3	x^4	x^5	x^6	x^7	x^8	x^9	x^{10}	x^{11}	x^{12}
1	1	1	0	0	0	0	0	1	0	0	0	1
1	1	1	0	0	0	0	0	0	0	1	0	1
1	0	1	0	0	0	0	1	1	0	0	0	1
1	0	0	1	1	0	1	0	0	1	1	0	1
1	0	0	1	0	0	1	1	1	0	0	1	1
1	1	0	1	1	0	0	0	0	1	0	1	1

[XV] Для пошуку примітивних поліномів над скінченими полями Галуа порядку 2^d — $GF(2^d)$ — можна скористатися функцією *Matlab* **gfprimfd**(**d**, **p**), де **d** — додатне ціле число; **p** — параметр, що може бути заданий як **'min'** чи **'max'** (при цьому повертається один примітивний поліном, що має, відповідно, мінімальну чи максимальну кількість ненульових коефіцієнтів), **'all'** (повертаються усі можливі примітивні поліноми), а також додатним цілим числом (тоді повертаються поліноми саме з **p** ненульовими коефіцієнтами).

ЛРЗЗЗ, що має **d** розрядів, реалізується програмним модулем (М.53), в якому початковий стан регістра (у десятковому представленні) визначається аргументом **s** — довільним цілим числом у межах від 1 до **A·Z**–1.

```
Vrnd(s) := | μ ← (1 1 0 0 0 0 0 1 0 0 0 1)^T                    (M.53)
           | R_dec_1 ← s ; R_bin ← D2B(R_dec_1 , d)
           | for i ∈ 2 .. 2^d − 1
           |   bit ← 0
           |   for j ∈ 1 .. d
           |     bit ← R_bin_j ⊕ bit  if μ_j
           |   R ← R_bin ; R_bin_1 ← bit
           |   for j ∈ 2 .. d
           |     R_bin_j ← R_j−1
           |   R_dec_i ← B2D(R_bin)
           | R_dec_2^d ← 2^d
           | R_dec
```

На початку модуля задається вектор **μ** вагових коефіцієнтів примітивного поліному $\rho(x)$ для елементів відвідної послідовності (для наочності вектор зображено як матрицю-рядок з наступним транспонуванням).

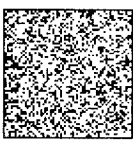
Циклом зміни індексу **i** проводиться перемикання станів регістра. Кожен **i**-й стан з двійкового формату конвертується до десяткового і зберігається у відповідному елементі вектора $\mathbf{R_{dec}}$. Оскільки період послідовності, що генерується даним регістром, дорівнює 2^d-1, а псевдовипадкова перестановка застосовуватиметься до вектора, кількість елементів в якому дорівнює $\mathbf{A{\cdot}Z} = 2^d$, в кінці (М.53) до сформованого вектора $\mathbf{R_{dec}}$ дописується ще один елемент, значення якого враховує верхній граничний індекс елементів вектора $\mathbf{W_{vec}}$.

Отримання масиву **Vrnd** розставлених у псевдовипадковому порядку індексів елементів вектора $\mathbf{W_{vec}}$ дозволяє провести генерування пар координат (по рядкам і стовпцям) кожного пікселя шляхом перетворення послідовності ПВЧ на двовимірну послідовність. Це, у свою чергу, робить можливим після псевдовипадкового обрання елемента з вектора $\mathbf{W_{vec}}$ помістити його значення до визначеного згенерованою парою координат елемента масиву, розмірність якого є ідентичною розмірності ЦВЗ.

Вищенаведена процедура реалізована у програмному модулі (М.54), де для кожного елемента *M*-послідовності **V** обчислюються індекси **a** і **z** елемента масиву $\mathbf{W}_{rnd}$, до якого заноситься поточний елемент вектора $\mathbf{W}_{vec}$. Функція **trunc**(*x*) повертає цілу частину від аргументу *x*, відкидаючи його мантису; функція **mod**(*k*, *m*) повертає залишок від ділення *k* на *m*. Додаванням одиниці враховано можливість повернення зазначеними функціями нульового результату.

$$W_{rnd} := \left| \begin{array}{l} V \leftarrow Vrnd(seed) \\ \text{for } i \in 1..A \cdot Z \\ \quad \left| \begin{array}{l} a \leftarrow \text{trunc}\left(\frac{V_i - 1}{A}\right) + 1 \\ z \leftarrow \text{mod}(V_i, Z) + 1 \\ W_{rnd_{a,z}} \leftarrow W_{vec_i} \end{array} \right. \\ W_{rnd} \end{array} \right. \qquad \text{(M.54)}$$

Приклад виконання (М.54) при **seed** := 23 наведено на рис. 5.41.



$W_{rnd} \cdot 255$

Рис. 5.41. Результат псевдовипадкової перестановки елементів ЦВЗ

Крок 6

Модуль розбиття масиву ЦВЗ на $\mathbf{n}_S$ блоків розмірністю **n**×**n** за своєю побудовою є аналогічним модулю (М.34). Відмінності полягають у наступному: змінна, якій присвоюється результат виконання модуля, позначається як $\mathbf{Б}_W$; виокремлення блоків проводиться з масиву $\mathbf{W}_{rnd}$; замість розмірності блока **N** використовується розмірність **n**; відповідно змінюється і граничне значення індексу рядка: замість **X** використовується **A**). Загальну кількість блоків, на яку розбивався контейнер ($\mathbf{N}_S$), з огляду на таку ж їх кількість у ЦВЗ, можна не змінювати.

Крок 7

Використовуючи програмні модулі (М.55) і (М.56), формуємо таблиці результатів сортування блоків контейнера (за значенням стан-

дартного відхилення елементів блоків) і блоків ЦВЗ (за кількістю значущих елементів).

$$T_C := \left| \begin{array}{l} \text{for } b \in 1..N_S \\ \quad \left| \begin{array}{l} T_{b,1} \leftarrow b \\ T_{b,2} \leftarrow \text{stdev}(Б_{C_b}) \end{array} \right. \\ \text{reverse}(\text{csort}(T,2)) \end{array} \right. \quad \text{(M.55)}$$

$$T_W := \left| \begin{array}{l} \text{for } b \in 1..n_S \\ \quad \left| \begin{array}{l} T_{b,1} \leftarrow b \\ T_{b,2} \leftarrow 0 \text{ on error } T_{b,2} \leftarrow \text{rows}(\text{match}(1, Б_{W_b})) \end{array} \right. \\ \text{reverse}(\text{csort}(T,2)) \end{array} \right. \quad \text{(M.56)}$$

До першого стовпця таблиць характеристик блоків контейнера (T_C) і ЦВЗ (T_W) вносяться порядкові індекси досліджуваних блоків. До другого — результат обчислення, відповідно, стандартного відхилення (функція **stdev**(...)) і кількості одиничних елементів (комплекс функцій **rows**(...) і **match**(...)).

Після формування таблиць, вони сортуються відповідно до значень другого стовпця (функція **csort**(T, 2)).

Фрагмент результату сортування для обраних у прикладі контейнера і ЦВЗ наведено у табл. 5.10.

Шляхом виокремлення перших стовпців з масивів T_C і T_W, зіставляємо індекси блоків контейнера з індексами блоків ЦВЗ:

$$T_\Sigma := \text{augment}(T_C^{\langle 1 \rangle}, T_W^{\langle 1 \rangle})$$

Крок 8

Згідно отриманого масиву поставлених у відповідність один одному індексів, проводимо перестановку блоків ЦВЗ у порядку, що відповідає даному зіставленню. Реалізацію даного етапу покладено на програмний модуль (М.57).

$$W_{sort} := \left| \begin{array}{l} \text{for } b \in 1..N_S \\ \quad W_{sort_{T_{\Sigma_{b,1}}}} \leftarrow Б_{W_{T_{\Sigma_{b,2}}}} \\ W_{sort} \end{array} \right. \quad \text{(M.57)}$$

У якості прикладу наведемо результат виконання (М.57) для другого рядка табл. 5.10:

$$W_{sort_{104}} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}; \qquad Б_{W_{104}} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

Таблиця 5.10

Приклад сортування блоків зображень контейнера і ЦВЗ

№ пор.	Індекси блоків контейнера	Значення стандартного відхилення	Індекси блоків ЦВЗ	Кількість значущих елементів
1	249	82.016	183	5
2	104	81.122	22	15
3	234	80.786	211	15
…	…	…	…	…
64	85	55.732	245	12
65	172	55.408	246	12
66	59	55.323	62	12
…	…	…	…	…
128	155	41.154	72	10
129	188	40.764	74	10
130	119	40.638	149	10
…	…	…	…	…
192	145	24.815	100	9
193	14	24.814	222	9
194	28	24.752	3	9
…	…	…	…	…
254	225	6.326	161	5
255	2	5.445	204	5
256	193	4.528	238	4

Крок 9

Використовуючи програмний модуль (М.43) (замість блоків $\mathbf{C_b}$ в якості аргументу функції **dct2d**(...) слід використати $\mathbf{Б_{C_b}}$), виконуємо пряме ДКП блоків зображення-контейнера.

З одержаних масивів коефіцієнтів ДКП $\mathbf{\Omega_b}$, що мають розмірності $\mathbf{N}\times\mathbf{N}$, необхідно виокремити лише СЧ-коефіцієнти (див. рис. 5.36), згорнувши їх до масивів $\mathbf{n}\times\mathbf{n}$.

Перед початком проведення власне виокремлення, формується масив координат виокремлюваних СЧ-коефіцієнтів — (М.58).

$$\Theta := \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 \\ 0 & 0 & 0 & 0 & 0 & 0 & 14 & 15 \\ 0 & 0 & 0 & 0 & 0 & 12 & 13 & 0 \\ 0 & 0 & 0 & 9 & 10 & 11 & 0 & 0 \\ 0 & 0 & 0 & 7 & 8 & 0 & 0 & 0 \\ 0 & 0 & 5 & 6 & 0 & 0 & 0 & 0 \\ 0 & 3 & 4 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

```
MF := | for d ∈ 1..n·n                                              (M.58)
      |   indices ← match(d, Θ)_1
      |   MF ← if(d = 1, indices, augment(MF, indices))
      | MF
```

У цьому випадку масив **MF** міститиме 16 стовпчиків, елементи кожного з яких нестимуть інформацію про індекси рядка і стовпця відповідного СЧ-коефіцієнта у масиві **Θ** (рис. 5.42).

MF =

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	⇐ *а)*
1	8	8	7	7	6	6	5	5	4	4	4	3	3	2	2	1	⇐ *б)*
2	1	2	2	3	3	4	4	5	4	5	6	6	7	7	8	8	⇐ *в)*

Рис. 5.42. Таблиця координат СЧ-коефіцієнтів ДКП:
а — порядкові №№ СЧ-коефіцієнтів;
б — індекси рядків масиву **Θ**;
в — індекси стовпців масиву **Θ**

У відповідності до таблиці з рис. 5.42, за допомогою програмного модуля (М.59) для кожного блока **b** виконується формування матриці Ω_{mid} обраних для модифікації коефіцієнтів. Результат виокремлення проілюстровано на рис. 5.43.

Крок 10

Для забезпечення можливості стеганографічного вбудовування ЦВЗ шляхом модифікації відношень між значеннями коефіцієнтів сусідніх блоків контейнера, попередньо задаємося таблицею квантування яскравості **Q** (М.60) і виокремлюємо з неї СЧ-коефіцієнти Q_{mid}, користуючись модулем, аналогічним (М.59), де замість Ω_{mid} використано Q_{mid}, а замість Ω_b — **Q**.

$$\Omega_{mid} := \left| \begin{array}{l} \text{for } b \in 1..N_S \\ \quad q \leftarrow 1 \\ \quad \text{for } \upsilon \in 1..n \\ \quad\quad \text{for } \nu \in 1..n \\ \quad\quad\quad \Omega_{mid\_b_{\upsilon,\nu}} \leftarrow \left| \Omega_b \right|_{MF_{1,q},MF_{2,q}} \\ \quad\quad\quad q \leftarrow q+1 \\ \quad \Omega_{mid_b} \leftarrow \Omega_{mid\_b} \\ \Omega_{mid} \end{array} \right. \qquad \text{(M.59)}$$

$$Q_{select} := \begin{array}{l} \bigcirc\ Qa \\ \bullet\ Qb \end{array} \qquad \text{(M.60)}^{\text{XVI}}$$

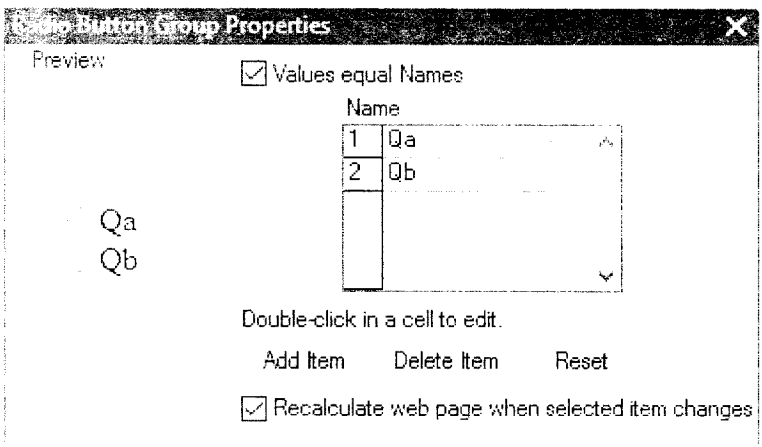
$$Q := \text{if}\left| Q_{select} = \text{"Qa"}, Qa, Qb \right|$$

Крок 11

Проводимо обчислення масивів полярностей блоків контейнера. Обрання виразу, за яким проводитиметься розрахунок, здійснюється програмним модулем (M.61):

$$\text{Expression} := \begin{array}{l} \bigcirc\ \text{вираз (5.29)} \\ \bullet\ \text{вираз (5.33)} \end{array} \qquad \text{(M.61)}$$

[XVI] Масиви **Qa** і **Qb** збігаються з таблицями рис. 5.38 *а* і *б* відповідно. Для реалізації механізму вибору використано мережний елемент інтерфейсу: (меню *Insert* ► ► *Control* ► *Web Control...* ► *Radio button group*):



$$\Omega_{104} = \begin{pmatrix} 997 & -387.54 & 183.54 & 251.46 & 6.25 & -48.72 & -17.41 & 9.53 \\ -44.76 & -261.7 & -280.85 & 45.09 & 43.82 & 34.94 & -22.53 & 11.64 \\ 16.17 & 121.77 & 28.78 & -22.72 & 27.51 & 18.19 & -3.89 & -13.99 \\ 53.25 & 44.21 & 50.96 & 29.72 & 3.51 & 25.07 & 4.54 & -35.4 \\ -28.75 & -2.19 & -10.64 & -21.97 & -2 & -8.35 & 18.75 & 17.91 \\ -14.29 & -39.41 & -14.57 & 1.68 & 18.36 & 4.76 & -6.43 & -2.27 \\ 12.25 & 27.76 & 10.61 & -12.18 & -13.29 & -2.85 & -1.28 & -9.1 \\ 2.53 & 6.75 & -1.51 & 14.04 & 9.67 & 7.4 & 2.25 & 2.73 \end{pmatrix}$$

$$\Omega_{mid_{104}} = \begin{pmatrix} 2.53 & 6.75 & 27.76 & 10.61 \\ -14.57 & 1.68 & -21.97 & -2 \\ 29.72 & 3.51 & 25.07 & 18.19 \\ -3.89 & -22.53 & 11.64 & 9.53 \end{pmatrix}$$

Рис. 5.43. Приклад виокремлення СЧ-коефіцієнтів з масиву Ω для 104-го блока контейнера

Формування масивів полярностей покладено на програмний модуль (М.62). При цьому, якщо за основу береться вираз (5.29), матриця СЧ-коефіцієнтів 1-го блока порівнюється з матрицею останнього (N_S) блока. У випадку ж реалізації виразу (5.33) функція **Trunc(x,y)** повертає результат обчислення виразу **trunc(x/y)·y**.

```
P := for b ∈ 1.. Ns                                                    (M.62)
       if Expression = "вираз (5.29)"
         Δ ← Ωmid_b − Ωmid_if(b="Ns", Ns, b−1)
       if Expression = "вираз (5.33)"
         Δ ← Trunc(|Ωmid_b|, Qmid) − trunc(|Ωb|_{·,·} / (ψ·Q_{·,·})) · Q_{·,·}
       Pb ← binarize(Δ, 0)
     P
```

Для блока № 104, наприклад, було отримано наступні масиви полярностей для виразів (5.29) і (5.33) (при $\psi = 250$):

$$P_{104}\ (5.29) = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \qquad P_{104}\ (5.33) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Крок 12

Згідно виразу (5.30), проводимо зміну поточних масивів полярності $\mathbf{P}_b$ у відповідності до елементів переставлених ЦВЗ $\mathbf{W}_{sort_b}$ — модуль (М.65).

$$\mathbf{P}^{\wedge} := \left| \begin{array}{l} \text{for } b \in 1..\, \mathbf{N_S} \\ \quad \mathbf{P}^{\wedge}_b \leftarrow \left| \mathbf{P}_b \oplus \mathbf{W}_{sort_b} \right| \\ \mathbf{P}^{\wedge} \end{array} \right. \qquad \text{(М.63)}$$

Для блока № 104, зокрема, для виразів (5.29) і (5.33) (в останньому випадку — при $\psi = 250$) було отримано такі результати:

$$\mathbf{P}^{\wedge}_{\cdot,104} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (5.29) \qquad \mathbf{P}^{\wedge}_{\cdot,104} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix} \quad (5.33)$$

Крок 13

На підставі поточних матриць СЧ-коефіцієнтів ($\mathbf{\Omega}_{mid_b}$) формуємо нові ($\mathbf{\Omega}^{\wedge}_{mid_b}$). Змін зазнають ті елементи первинної матриці, за координатами яких виконується нерівність виду $(\mathbf{P}^{\wedge}_b)_{\nu,\nu} \neq (\mathbf{P}_b)_{\nu,\nu}$ — див. програмний модуль (М.64). Модифікація проводиться таким чином, щоб при виконанні програмного модуля (М.62) можна було одержати масив $\mathbf{P}^*$, ідентичний масиву $\mathbf{P}^{\wedge}$, якщо при обчисленні параметра Δ в якості зменшуваного виступатиме елемент матриці $\mathbf{\Omega}^{\wedge}_{mid_b}$.

Наприклад, при $\psi = 250$ та $\eta = 25$ для блока № 104 маємо

$$\mathbf{\Omega}^{\wedge}_{mid_{\cdot,104}} = \begin{pmatrix} -34.98 & 35.59 & -41.72 & 40.1 \\ 19.77 & 38.01 & 51.49 & -62.62 \\ -88.96 & -62.93 & -9.33 & -53.76 \\ -73.15 & -22.53 & -30.75 & -36.44 \end{pmatrix} \quad (5.29) \qquad \mathbf{\Omega}^{\wedge}_{mid_{\cdot,104}} = \begin{pmatrix} 52.53 & 56.75 & 52.76 & 60.61 \\ -39.57 & 51.68 & -46.97 & -52 \\ 4.72 & 28.51 & 50.07 & 43.19 \\ -53.89 & -22.53 & 36.64 & 34.53 \end{pmatrix} \quad (5.33)$$

Крок 14

Модифіковані для кожного блока матриці СЧ-коефіцієнтів ($\mathbf{\Omega}_{mid_b}$) відображуються до загальних матриць коефіцієнтів ДКП ($\mathbf{\Omega}^{\wedge}_b$) — програмний модуль (М.65). При цьому так само використовується таблиця координат середньочастотних коефіцієнтів ДКП **MF** (див. (М.58) і рис. 5.42).

```
Ω^mid := | for  b ∈ 1..N_S
         |   | Ω^mid_b ← Ωmid_b
         |   | for  υ ∈ 1..n
         |   |   for  ν ∈ 1..n
         |   |     if (P^_b)_{υ,ν} ≠ (P_b)_{υ,ν}
         |   |       | if Expression = 'вираз (5.29)'
         |   |       |   (Ω^mid_b)_{υ,ν} ← (Ωmid_{if(b=1, N_S, b−1)})_{υ,ν} + if[(P^_b)_{υ,ν}, η, −η]
         |   |       | if Expression = 'вираз (5.33)'
         |   |       |   | if (P^_b)_{υ,ν} = 0
         |   |       |   |   while Trunc[|(Ω^mid_b)_{υ,ν}|, Qmid_{υ,ν}] > Trunc[(Ω_b)_{1,1}/ψ, Q_{1,1}]
         |   |       |   |     (Ω^mid_b)_{υ,ν} ← (Ω^mid_b)_{υ,ν} + if[(Ω^mid_b)_{υ,ν} ≤ 0, η, −η]
         |   |       |   | if (P^_b)_{υ,ν} = 1
         |   |       |   |   while Trunc[|(Ω^mid_b)_{υ,ν}|, Qmid_{υ,ν}] ≤ Trunc[(Ω_b)_{1,1}/ψ, Q_{1,1}]
         |   |       |   |     (Ω^mid_b)_{υ,ν} ← (Ω^mid_b)_{υ,ν} + if[(Ω^mid_b)_{υ,ν} ≥ 0, η, −η]
         | Ω^mid
```
(M.64)

```
Ω^ := | for  b ∈ 1..N_S
      |   | Ω^_b ← Ω_b
      |   | q ← 1
      |   | for  υ ∈ 1..n
      |   |   for  ν ∈ 1..n
      |   |     | (Ω^_b)_{MF_{1,q}, MF_{2,q}} ← (Ω^mid_b)_{υ,ν}
      |   |     | q ← q + 1
      | Ω^
```
(M.65)

Крок 15

До матриць-контейнерів $\Omega^{\wedge}{}_{b}$ необхідно застосувати зворотне ДКП (програмний модуль (М.66)) і сформувати на основі **N_S** блоків **$Б^{\wedge}{}_{b}$** загальний масив **S** контейнера-результату (програмний модуль (М.67)).

```
Б^ := | for  b ∈ 1..N_S
      |     Б^_b ← idct2d(Ω^_b)
      | Б^
```
(M.66)

$$
\begin{aligned}
S := \Big| & \text{for } b \in 1..\frac{X}{N} \\
& \quad S \leftarrow \text{if}(b = 1, Б^{\wedge}_{1}, \text{stack}(S, Б^{\wedge}_{b})) \\
& S' \leftarrow 0 \\
& \text{for } b \in \frac{X}{N} + 1..N_S \\
& \quad S' \leftarrow \text{if}(S' = 0, Б^{\wedge}_{b}, \text{stack}(S', Б^{\wedge}_{b})) \\
& \quad \text{if } \text{mod}\left(b, \frac{X}{N}\right) = 0 \\
& \quad\quad S \leftarrow \text{augment}(S, S') \\
& \quad\quad S' \leftarrow 0 \\
& \text{clip}(S, 0, 255)
\end{aligned}
\qquad \text{(М.67)}
$$

Отримане при цьому зображення із вбудованим ЦВЗ за певних обставин може занадто втратити у яскравості, що викликано декількома причинами: по-перше, для спрощення програмних модулів не була проведена оптимізація вбудовування за формулою (5.31); по-друге, сусідні блоки контейнера можуть мати досить різні значення інтенсивностей і, відповідно, СЧ-коефіцієнтів ДКП, що викликає необхідність при побудові алгоритму за (5.29) істотно змінювати значення цих коефіцієнтів для задоволення поставлених умов. У комплексі ці дві причини викликають появу пікселів контейнера, яскравість яких після проведення зворотного ДКП, виходить за межі [0, 255]. Останнє враховується обрізкою недопустимих значень наприкінці (М.67) за допомогою вбудованої функції **clip**(...).

Результати вбудовування ЦВЗ до контейнера шляхом модифікації відношення між значеннями коефіцієнтів сусідніх блоків і в межах блока представлені на рис. 5.44.

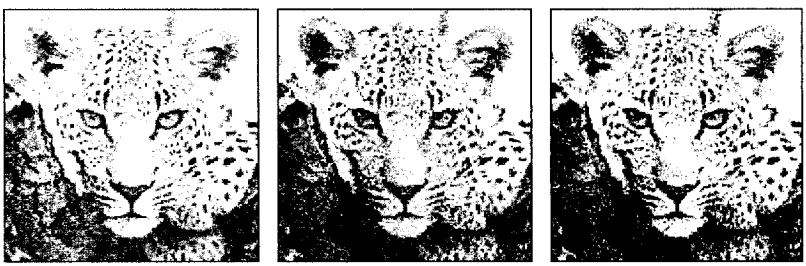


Рис. 5.44. Контейнер-оригінал (*а*) і контейнери з ЦВЗ, вбудованими згідно виразів (5.29) (*б*) і (5.33) (*в*)

Крок 16

Розглянемо процес видобування ЦВЗ з зображення-контейнера. Як було зазначено авторами [104], для видобування ЦВЗ окрім, власне, контейнера з, можливо, вбудованим ЦВЗ (**S***) є необхідною ще й наявність контейнера-оригіналу (**C**) і зображення ЦВЗ (**W**).

Зображення **C** і **S*** розбиваємо на блоки $Б_C$ і $Б^*_S$ відповідно, використовуючи програмний модуль (М.34), з належною заміною **B** на **C** або **S***.

До кожного блока оригінального зображення і зображення, досліджуваного на ЦВЗ, застосовуємо пряме ДКП (модуль (М.43)), а на основі одержаних при цьому матриць коефіцієнтів (Ω і Ω^*) формуємо матриці СЧ-коефіцієнтів Ω_{mid} і Ω^*_{mid} (модуль (М.59)), що їх надалі ми використовуватимемо для обчислення шаблонів полярності **P** і **P*** відповідно (модуль (М.62)).

Шляхом поблокового додавання за модулем 2 одержаних матриць полярностей, одержуємо двійкові дані, які, у тому випадку, якщо контейнер дійсно містить ЦВЗ, відповідають переставленим у просторі і псевдовипадково змішаним елементам ЦВЗ. Операцію додавання виконуємо за допомогою програмного модуля (М.68).

$$W^*_{sort} := \left| \begin{array}{l} \text{for } b \in 1..N^*_S \\ \quad W^*_{sort_b} \leftarrow \overrightarrow{\left| P_b \oplus P^*_b \right|} \\ W^*_{sort} \end{array} \right. \qquad \text{(М.68)}$$

Формуємо масив T_Σ індексів співставлених пар блоків контейнера і оригінального ЦВЗ (див. (М.55), (М.56)), на підставі якого виконуємо зворотну просторову перестановку блоків масиву W^*_{sort} — програмний модуль (М.69).

$$Б^*_W := \left| \begin{array}{l} \text{for } b \in 1..N^*_S \\ \quad \left| \begin{array}{l} i \leftarrow T_{\Sigma_{b,1}},\ j \leftarrow T_{\Sigma_{b,2}} \\ Б^*_{W_j} \leftarrow W^*_{sort_i} \end{array} \right. \\ Б^*_W \end{array} \right. \qquad \text{(М.69)}$$

Переставлені у просторі блоки $Б^*_W$ об'єднуємо у спільний масив W^*_{rnd}, елементи якого, гіпотетично, є псевдовипадково змішаними елементами оригінального ЦВЗ — програмний модуль (М.70).

$$W^*_{rnd} := \left| \begin{array}{l} \text{for } b \in 1..\frac{A}{n} \\ \quad W^*_{rnd} \leftarrow \text{if}\left(b = 1, Б^*_{w_s}, \text{stack}\left(W^*_{rnd}, Б^*_{w_b}\right)\right) \\ W'_{rnd} \leftarrow 0 \\ \text{for } b \in \frac{A}{n}+1..n^*_S \\ \quad W'_{rnd} \leftarrow \text{if}\left(W'_{rnd} = 0, Б^*_{w_b}, \text{stack}\left(W'_{rnd}, Б^*_{w_b}\right)\right) \\ \quad \text{if } \text{mod}\left(b, \frac{A}{n}\right) = 0 \\ \quad\quad W^*_{rnd} \leftarrow \text{augment}\left(W^*_{rnd}, W'_{rnd}\right) \\ \quad\quad W'_{rnd} \leftarrow 0 \\ W^*_{rnd} \end{array} \right. \tag{M.70}$$

Використовуючи програмний модуль (М.71), проводимо зворотну псевдовипадкову перестановку видобутих даних. При цьому застосовуємо (М.53) для одержання ПВЧ, на основі яких генерується ідентична пара координат елемента в масиві W^*_{rnd}, значення якого, у свою чергу, присвоюється i-му елементові вектора W^*_{vec}.

$$W^*_{vec} := \left| \begin{array}{l} V \leftarrow Vrnd(seed^*) \\ \text{for } i \in 1..A^* \cdot Z^* \\ \quad a \leftarrow \text{trunc}\left(\frac{V_i - 1}{A^*}\right) + 1 \\ \quad z \leftarrow \text{mod}\left(V_i, Z\right) + 1 \\ \quad W^*_{vec_i} \leftarrow W^*_{rnd_{a,z}} \\ W^*_{vec} \end{array} \right. \tag{M.71}$$

Одержаний у результаті виконання (М.71) вектор згортаємо до масиву W^* з розмірністю оригінального ЦВЗ (модуль (М.72)).

$$W^* := \left| \begin{array}{l} \text{for } z \in 1..Z^* \\ \quad W^{*\langle z \rangle} \leftarrow \text{submatrix}\left[W^*_{vec}, (z-1)\cdot A^*+1, z\cdot A^*, 1, 1\right] \\ W^* \end{array} \right. \tag{M.72}$$

Графічне представлення видобутих ЦВЗ, вбудовування яких до контейнера було проведене шляхом зміни відношень між значеннями коефіцієнтів ДКП сусідніх блоків і в межах одного блока при $\psi = 250$ та $\eta = 25$ зображено на рис. 5.45.

а) *б)*

Рис. 5.45. ЦВЗ, видобуті з контейнера **S**, заповненого згідно виразів (5.29) (*а*) і (5.33) (*б*)

Крок 17

Результати обчислення показників візуального спотворення для двох розглянутих різновидів методу при $\psi = 250$ та $\eta = 25$ занесені до табл. 5.11 (стор. 234). Зауважимо, що при порівнянні отриманих результатів з результатами інших методів, слід брати до уваги, що до контейнера вбудовувалася інформація, об'єм якої (у пікселях) був усього лишень у 4 рази меншим за ємність контейнера.

Крок 18

Результати видобування ЦВЗ **W** зі стиснутого із втратами контейнера наведені на рис. 5.46, 5.47.

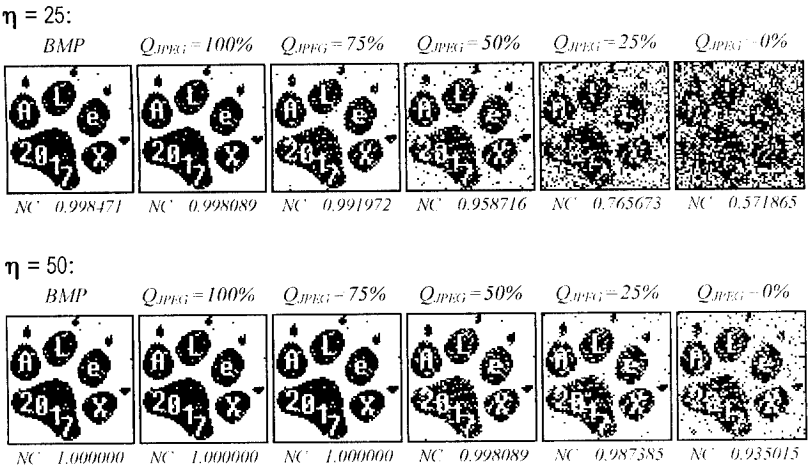
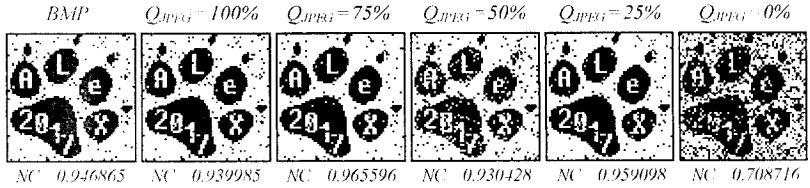


Рис. 5.46. ЦВЗ, видобуті зі стиснутого за алгоритмом *JPEG* контейнера **S**, заповненого згідно виразу (5.29)

$\psi = 60$, $\eta = 25$:



$\psi = 250$, $\eta = 25$:

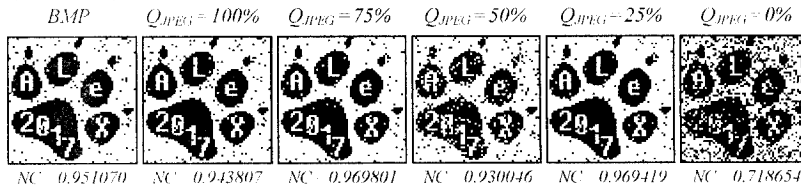


Рис. 5.47. ЦВЗ, видобуті зі стиснутого за алгоритмом *JPEG* контейнера **S**, заповненого згідно виразу (5.33)

Крок 19

Результати видобування ЦВЗ **W** з контейнера, обробленого ґаусівським розмиттям (*Gaussian blur*) з різним стандартним відхиленням σ, наведені на рис. 5.48, 5.49.

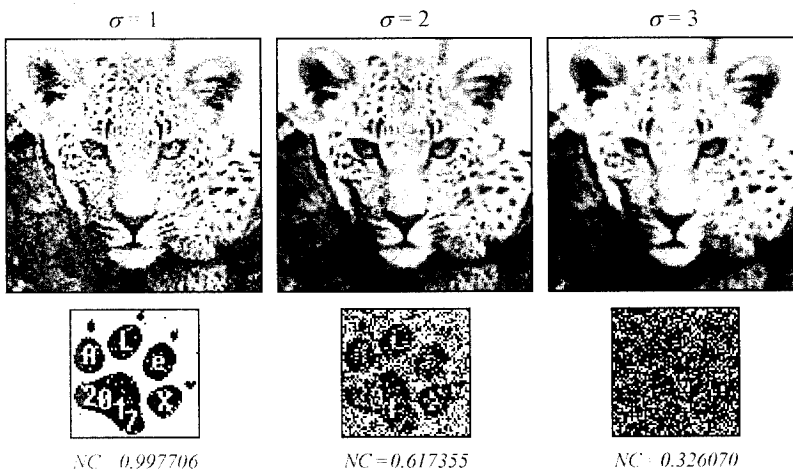


Рис. 5.48. Результат видобування ЦВЗ з розмитого контейнера **S**, заповненого за виразом (5.29) при $\eta = 25$

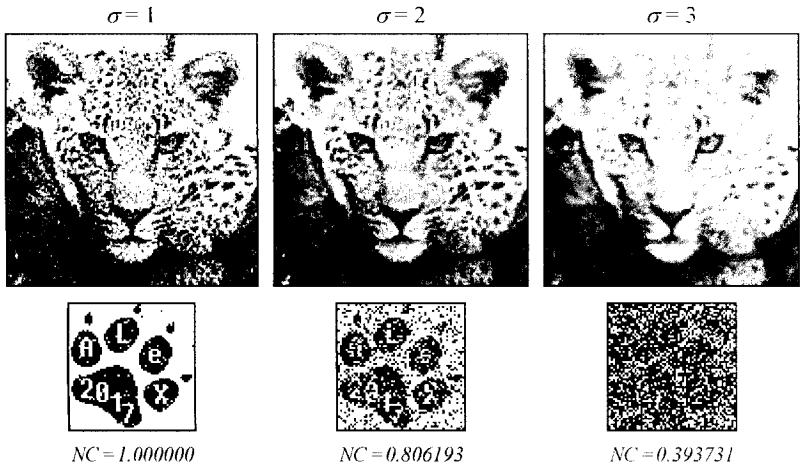


Рис. 5.49. Результат видобування ЦВЗ з розмитого контейнера **S**, заповненого за виразом (5.29) при η = 50

Крок 20

Результати впливу на якість видобутого ЦВЗ операції збільшення контрастності контейнера наведені на рис. 5.50.

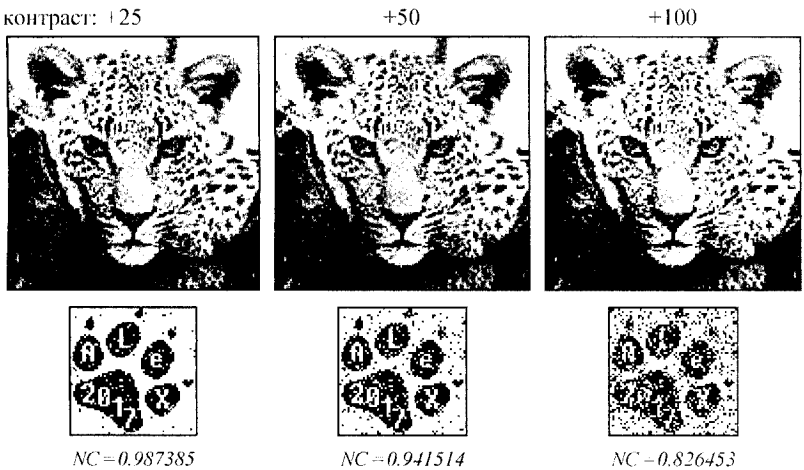


Рис. 5.50. Результат видобування ЦВЗ з контейнера **S**, заповненого за виразом (5.29) при η = 25, при збільшенні контрастності останнього

Крок 21

Вплив на якість видобутого ЦВЗ операцій зашумлення та кадрування контейнера проілюстрований, відповідно, на рис. 5.51, 5.52.

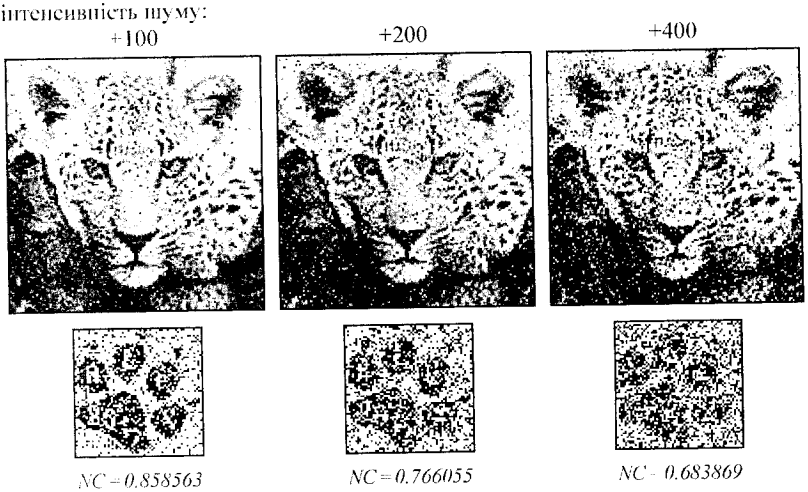


Рис. 5.51. Результат видобування ЦВЗ з контейнера **S**, заповненого за виразом (5.29) при $\eta = 25$, при зашумленні останнього

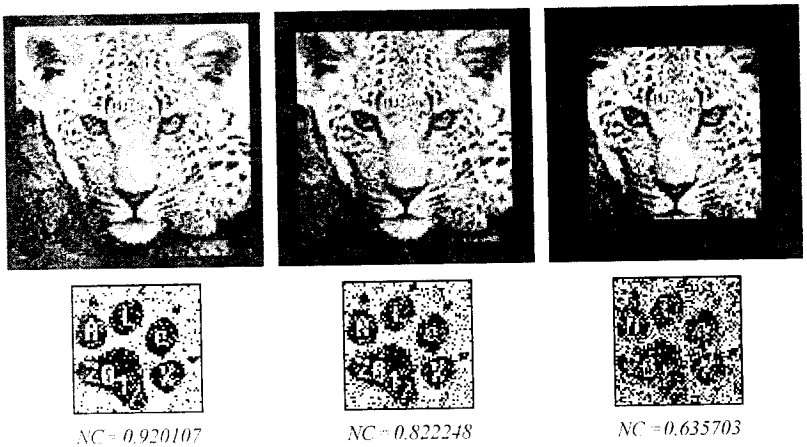


Рис. 5.52. Результат видобування ЦВЗ з контейнера **S**, заповненого за виразом (5.29) при $\eta = 25$, при кадруванні останнього

5.3.3.4. Метод Фрідріх

Метод стеганографічного заповнення графічного контейнера, запропонований у [105] Джессікою Фрідріх (*J. Fridrich*), є комбінацією двох алгоритмів: за одним з них приховувані дані вбудовуються до низькочастотних, а за іншим — до середньочастотних коефіцієнтів ДКП сегментів контейнера. Як було показано автором, таке каскадне використання двох кардинально різних за впливом на контейнер підходів дозволяє отримати непогані результати стосовно стійкості стеганосистеми до атак.

Зображення, яке планується використати в якості контейнера, спочатку конвертується у сигнал з нульовим математичним сподіванням і певним стандартним відхиленням таким чином, щоб НЧ-коефіцієнти ДКП, які будуть одержані в подальшому, потрапляли у попередньо заданий незмінний діапазон. Запропоноване Дж. Фрідріх перетворення

$$\mathbf{G} = \frac{1024}{\sqrt{X \cdot Y}} \cdot \frac{\mathbf{C} - \bar{\mathbf{C}}}{\sigma(\mathbf{C})}, \tag{5.37}$$

де X, Y — розмірність зображення $\mathbf{C}$ у пікселях; $\bar{\mathbf{C}}$ і $\sigma(\mathbf{C})$ — відповідно, математичне сподівання і стандартне відхилення значень яскравості пікселів зображення, — трансформує півтонове зображення $\mathbf{C}$ у двовимірний сигнал $\mathbf{G}$ з нульовим математичним сподіванням, такий, що абсолютне значення максимального НЧ-коефіцієнта ДКП сигналу $\mathbf{G}$ не перевищуватиме поріг у 200...250 одиниць. При цьому стверджується, що дане перетворення є застосовним для широкого кола різноманітних зображень: як з великими однорідними ділянками, так і доволі текстурованих.

Для зображення $\mathbf{G}$ виконується обчислення коефіцієнтів ДКП, з яких модифікуються лише низькочастотні. Причому зміна має проводитися таким чином, щоб у цих коефіцієнтах було закодовано сигнал прихованого повідомлення $\mathbf{W}$, що являє собою послідовність чисел $\{-1, 1\}$. Для цього попередньо необхідно визначити геометричну прогресію дійсних чисел:

$$\tau_{i+1} = \frac{1+\alpha}{1-\alpha} \cdot \tau_i; \quad \tau_1 = 1, \tag{5.38}$$

параметризовувану (налаштовувану) за допомогою параметра $\alpha \in (0, 1)$.

Для значень $t > 1$, $\tau_i \le t < \tau_{i+1}$ визначається індексна функція

$$ind(t) = (-1)^i, \text{ якщо } t \in [\tau_i, \tau_{i-1}), \tag{5.39}$$

що для кожного дійсного числа $t > 1$ дозволяє визначити його індекс (± 1). Цілком очевидно, що цей індекс може бути змінений шляхом додавання чи віднімання числа, яке не перевищує значення $\alpha \cdot t$. Поведінку індексних функцій для $\alpha = 0.1$, 0.2 і 0.3 проілюстровано на рис. 5.53.

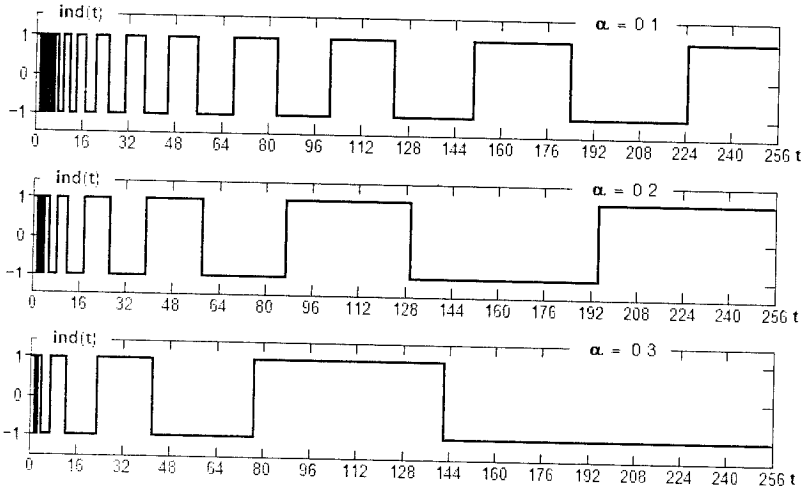


Рис. 5.53. Індексні функції $ind(t)$ при значеннях $\alpha = 0.1$, 0.2 і 0.3.

Для вбудовування масиву повідомлення $\mathbf{W}$, кожен з N_W бітів якого може приймати значення $\mathbf{W}_j \in \{-1, 1\}$, де $j \in \{1, 2, \ldots, N_W\}$, обираються $N_{\Omega\_LF} = N_W$ НЧ-коефіцієнтів ДКП — Ω_j, значення яких змінюються таким чином, щоб задовольнялася умова $ind\left(\left|\Omega\hat{}_j\right|\right) = W_j$, де $\Omega\hat{}_j$ — модифіковане значення коефіцієнта ДКП. У тому випадку, якщо $|\Omega_j| < 1$, коефіцієнт ДКП для перенесення елемента стеганограми не використовується.

Завдяки властивостям індексної функції, як зазначається у [105], кожен коефіцієнт може бути змінений не більше ніж на $100 \cdot \alpha$ відсотків. Більш точно максимально можливу зміну, яку може зазнати коефіцієнт ДКП, можна обчислити, розглянувши граничний випадок, коли первинне значення коефіцієнта потрапляє на інтервал $\left[\tau_b, \tau_{b+1}\right)$, якнайближче до τ_{b-1}, (див. рис. 5.54, значення індексної функції умовні). Також припустимо, що внаслідок випадкового обрання з інтервалу

$\left[\tau_{b-1}, \tau_b\right)$ (вважаємо, що заміна відбувається у бік менших значень), первинному значенню коефіцієнта було присвоєно значення τ_{b-1}. При цьому наближене до τ_{b+1} значення $\left(\tau^{\approx}_{b+1}\right)$ перевищує τ_{b-1} у $\left(\frac{1+\alpha}{1-\alpha}\right)^2$ разів або на $\psi(\alpha)=\left[\left(\frac{1+\alpha}{1-\alpha}\right)^2-1\right]\cdot 100\Big/\left(\frac{1+\alpha}{1-\alpha}\right)^2$ відсотків. Графік залежності максимально можливої зміни коефіцієнта ДКП від значень параметра α наведено на рис. 5.55.

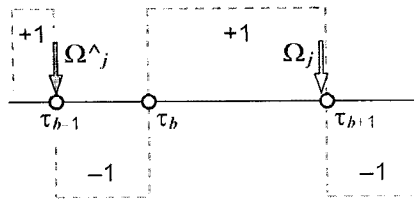


Рис. 5.54. До пояснення визначення максимально можливої зміни, що її може зазнати окремий коефіцієнт ДКП

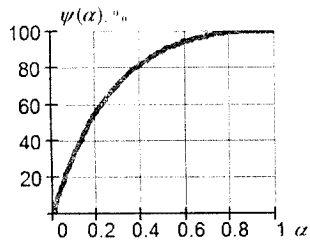


Рис. 5.55. Залежність максимально можливої зміни коефіцієнта ДКП від значення α

Також у [105] зазначається, що зміни коефіцієнтів ДКП носитимуть випадковий характер, оскільки не існує жодних підстав вважати, що на початковому етапі кодування вони є результатом вбудовування певного повідомлення.

Найбільша стійкість стеганосистеми до спотворювань контейнера досягається за встановлення в якості нових значень коефіцієнтів ДКП середини інтервалів $\left[\tau_i, \tau_{i+1}\right)$. Проте, це може посприяти появі скупчень

однакових коефіцієнтів, що, у свою чергу, зробить таку систему ненадійною з точки зору можливого стеганоаналізу. Зазвичай значення параметра α обирається таким, щоб вбудовування повідомлення до контейнера не призводило до помітних оку спотворень останнього.

Операція видобування проводиться шляхом виконання аналогічних до операції вбудовування перетворень контейнера, в якому, як вважається, є наявним вбудоване повідомлення:

- конвертація у сигнал з нульовим математичним сподіванням за формулою (5.37);
- розрахунок коефіцієнтів ДКП конвертованого зображення;
- обчислення для заздалегідь оговорених коефіцієнтів ДКП індексної функції (5.39) при заданому параметрі α;
- формування з одержаних індексів масиву видобутої копії повідомлення W.

Крім того, у [105] пропонується ще й метод детектування наявності/відсутності вбудованого повідомлення у контейнері, що може бути корисним при захищенні цифрового контенту (інформаційного змісту) за допомогою ЦВЗ. Дана операція передбачає поінформованість одержувача стосовно змісту прихованого повідомлення.

Оскільки через вбудовування більшість з N_{Ω_LF} НЧ-коефіцієнтів зазнала модифікації, просте обчислення кореляції між W_j та $ind\left(\left|\Omega\hat{}_j\right|\right)$ зумовлювало б собою нестійкість методу, позаяк малі, візуально незначущі коефіцієнти ДКП робитимуть до загальної енергії сигналу внесок тієї ж ваги, що й великі, візуально більш значущі коефіцієнти.

Разом з тим, зважаючи на попередньо поставлену умову, що контейнер з вбудованим повідомленням не повинен привертати увагу, вбудовувати дані лише до коефіцієнтів, що мають велике значення, ми теж не можемо. Крім того, позиції найбільших коефіцієнтів ДКП первинного і модифікованого зображень можуть не збігатися, що унеможливить безпомилкову ідентифікацію тих з них, до яких було проведено вбудовування. У пропонованій Дж. Фрідріх системі вбудовування відбувається до всіх НЧ-коефіцієнтів, незалежно від їхнього значення (крім тих, звичайно, що є меншими за одиницю), але тільки найбільші з них враховуються згодом при обчисленні коефіцієнта кореляції, зважуваного з енергією абсолютних значень коефіцієнтів ДКП:

$$K = \sum_{j=1}^{N_{\Omega\_LF}} \left|\Omega_j^*\right|^{\beta} \cdot ind\left(\left|\Omega_j^*\right|\right) \cdot W_j \Big/ \sum_{j=1}^{N_{\Omega\_LF}} \left|\Omega_j^*\right|^{\beta}. \qquad (5.40)$$

Таке зважування автоматично робить більш виразними найбільші значення коефіцієнтів, одночасно пригнічуючи незначні, які могли б зазнати змін у результаті певних операцій з обробки зображення.

Параметр β у (5.40) встановлює важливість зважування. Якщо $\beta = 0$, обчислюється звичайний, незважений коефіцієнт кореляції. Значення β, надто наближене до 1, призводить до сингулярності (виродження) системи детектування: функція виявлення залежатиме тільки від значення лишень одного біта, що відповідає максимальному коефіцієнтові ДКП. Автор методу рекомендує використовувати $\beta \in (0.5, 1)$.

Більш стійкою до атак дану систему можна зробити шляхом пошуку максимального значення коефіцієнта кореляції відносно стандартного відхилення значень яскравості пікселів зображення, підозрюваного на наявність вбудованого повідомлення.

Результат масштабування (5.37) залежить від стандартного відхилення значень яскравості пікселів, що може бути суттєво спотворене, якщо зображення з вбудованим повідомленням попередньо зазнало згладжування або зашумлення. Як наслідок, коефіцієнти ДКП такого зображення будуть промасштабовані за допомогою фіксованого коефіцієнта (відношення стандартних відхилень оригінального і досліджуваного на наявність прихованого повідомлення зображень:

$$d = \sigma(\mathbf{C}) / \sigma(\mathbf{S}).$$

Проте, повідомлення, закодованого у коефіцієнтах ДКП, лінійні зміни не торкнуться. Останнє й наводить на думку про доцільність використання простого одновимірного пошуку правильного масштабу d, що максимізував би значення коефіцієнта кореляції (оскільки, як було зазначено вище, копія первинного зображення-контейнера у детекторі відсутня). Таким чином, доповнена функція детектування має наступний вигляд:

$$K' = \max_{d \in (1-\delta,\, 1+\delta)} K(d) = \sum_{j=1}^{N_{\Omega\,LF}} \left|\Omega_j^*\right|^{\beta} \cdot ind\left(d \cdot \left|\Omega_j^*\right|\right) \cdot W_j \Bigg/ \sum_{j=1}^{N_{\Omega\,LF}} \left|\Omega_j^*\right|^{\beta}. \quad (5.41)$$

Автором [105] було встановлено, що навіть у випадку значних спотворень зображення в результаті атак, достатнім є крок $\delta = 0.25$.

Ускладнення детектування, що при цьому виникають, вимагають скорочення інформаційної частини повідомлення і додавання корегувальних бітів. Таким чином, оскільки внесок у виявлення повідомлення роблять лише найбільші коефіцієнти ДКП, інформаційний зміст повідомлення довжиною N_W становить собою лише певну частину від N_W. Крім того, є цілком очевидним, що одновимірний пошук масштабного

коефіцієнта, який би максимізував коефіцієнт кореляції, збільшуватиме відсоток помилкових виявлень.

Для забезпечення високої стійкості до атак на стеганосистему при найменшому (наскільки, звичайно ж, це є можливим) спотворенні контейнера, автором методу було запропоновано вбудувати до останнього додаткове повідомлення, використовуючи методику розширення спектра. При цьому вбудовування відбувається шляхом додавання шумоподібного сигналу до СЧ-коефіцієнтів ДКП зображення (кількість N_{Ω_MF} яких становить близько 30% від загальної кількості коефіцієнтів ДКП).

Вважатимемо, що інформація, вміщувана у додатковому повідомленні, складається з N_{W+} символів $\mathbf{W}_j^+$, кожен з яких може бути представлений десятковим цілим числом, $1 \le \mathbf{W}_j^+ \le \boldsymbol{max}(\mathbf{W}^+)$. Для кожного j-го символу генерується послідовність $\boldsymbol{\xi}^{(j)}$ ПВЧ, рівномірно розподілених на інтервалі [0, 1]. Початковий стан генератора ПВЧ може виступати в ролі секретного ключа. Потужність j-ї множини ПВЧ: $\left|\boldsymbol{\xi}^{(j)}\right| \ge N_{\Omega\_MF} + \boldsymbol{max}(\mathbf{W}^+)$. Для представлення окремого символу повідомлення $\mathbf{W}^+$, з множини ПВЧ $\boldsymbol{\xi}^{(j)}$ виокремлюється сегмент $\boldsymbol{\eta}^{(j)} = \xi_{W_j^+}^{(j)}, \ldots, \xi_{W_j^+ + N_{\Omega\_MF}-1}^{(j)}$, який містить N_{Ω_MF} елементів. У підсумку, повідомлення з N_{W+} символів може бути представлене у вигляді наступної зваженої суми[XVII]:

$$\boldsymbol{Spr} = \frac{\left[\sum_{j=1}^{N_{W^+}} \boldsymbol{\eta}^{(j)}\right] - \frac{N_{W^+}}{2}}{\sqrt{N_{W^+}/12}}. \qquad (5.42)$$

Сигнал з розширеним спектром ***Spr*** має приблизно нормальний (гаусівський) розподіл з нульовим математичним сподіванням і одиничним стандартним відхиленням (точність апроксимації зростає зі збільшенням значення N_{W+}). У подальшому сигнал ***Spr*** помножується на параметр γ, що, будучи поелементно доданим до N_{Ω_MF} обраних

[XVII] Слід зазначити, що для формування нормального розподілу на основі рівномірного у [105] пропонується використати вираз $\boldsymbol{Spr} = \frac{1}{\sqrt{N_{W^+}}} \cdot \sum_{j=1}^{N_{W^+}} \boldsymbol{\eta}^{(j)}$, що, як відомо з теорії статистичних розподілів, не зовсім відповідає поставленим вимогам (див., наприклад, [106]).

СЧ-коефіцієнтів, регулює відношення «стійкість/помітність вбудовування».

Видобування повідомлення $\mathbf{W}^+$ проводиться шляхом попереднього обчислення коефіцієнтів ДКП зображення й виокремлення серед них саме середньочастотних (дана операція повинна бути узгодженою з відповідною дією на етапі заповнення контейнера). Використовуючи секретний ключ/алгоритм, виконується генерування послідовностей ПВЧ (загальною кількістю N_{W^+}, якщо даний параметр є відомим; в іншому випадку — за ситуацією, виходячи з аналізу вже видобутої частини повідомлення) довжиною $N_{\Omega\_MF} + \max(\mathbf{W}^+)$.

З кожної послідовності ПВЧ $\xi^{(j)}$ виокремлюється $\max(\mathbf{W}^+)$ сегментів довжиною N_{Ω_MF} елементів, для яких обчислюється взаємна кореляція з вектором виокремлених СЧ-коефіцієнтів. Позиція найбільшого значення кореляції в одержаному при цьому векторі і визначатиме собою значення, яке мав вбудований символ $\mathbf{W}_j^+$.

Перейдемо до розгляду практичної реалізації даного методу.

Крок 1

Нехай зображення-контейнер і зображення-ЦВЗ представлені графічними файлами C.bmp і W.bmp відповідно (рис. 5.56):

$$\mathbf{C} := \mathbf{READBMP}(\text{"C.bmp"}); \quad \mathbf{W} := \mathbf{READBMP}(\text{"W.bmp"}).$$

При цьому характеристики, що відповідають зазначеним зображенням, є наступними:

$$\mathbf{X} := \mathbf{rows}(\mathbf{C}) = 256 \text{ пікс.}; \quad \mathbf{Y} := \mathbf{cols}(\mathbf{C}) = 256 \text{ пікс.};$$
$$\mathbf{A} := \mathbf{rows}(\mathbf{W}) = 32 \text{ пікс.}; \quad \mathbf{Z} := \mathbf{cols}(\mathbf{W}) = 32 \text{ пікс.}$$

Крок 2

Для більшої стійкості стеганосистеми, розіб'ємо контейнер на блоки $\mathbf{Б_C}$ розмірністю $\mathbf{N}\times\mathbf{N}$, де $\mathbf{N} := 128$ пікс. (див. програмний модуль (М.34)), до кожного з яких вбудовуватимемо один і той самий ЦВЗ. Кількість отримуваних за цих умов сегментів — $\mathbf{N_S} := \mathbf{X}\cdot\mathbf{Y} \div \mathbf{N}^2 = 4$.

Крок 3

Використовуючи програмний модуль (М.73), для кожного блока $\mathbf{Б_C}$ реалізуємо трансформацію (5.37), враховуючи, що розмірність блоків — $\mathbf{N}\times\mathbf{N}$ пікселів.

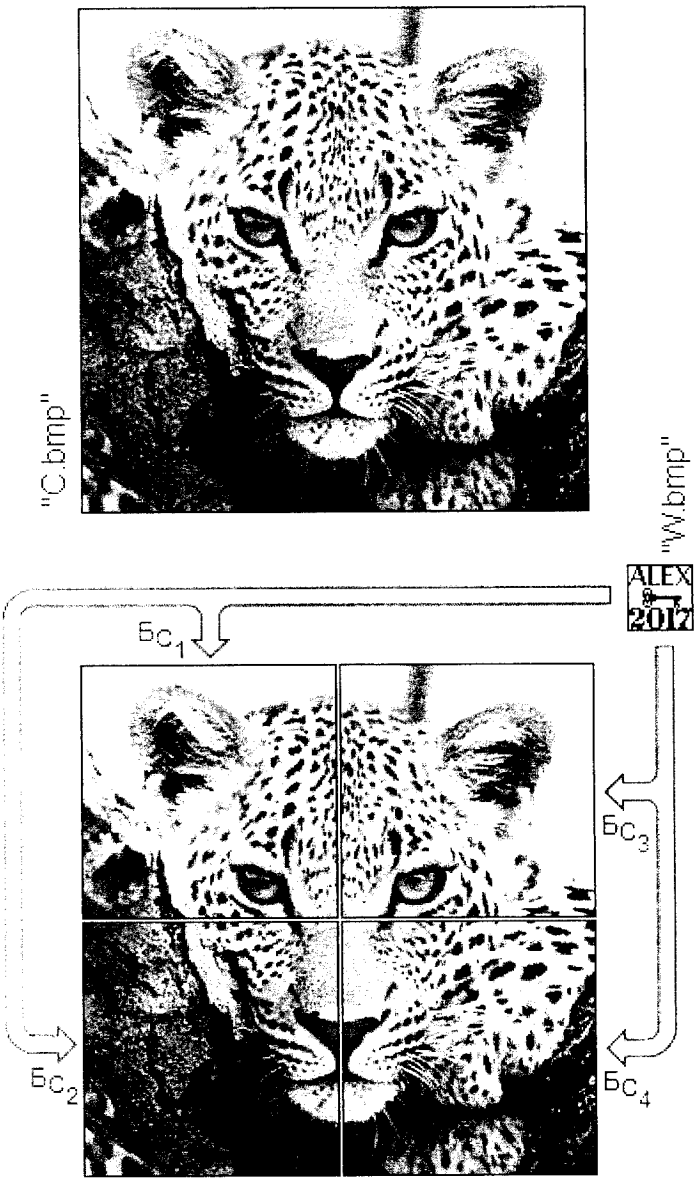


Рис. 5.56. Приклад контейнера-оригіналу ("C.bmp"), ЦВЗ ("W.bmp") і контейнера, розбитого на $\mathbf{N_S}$ = 4 блоки ($\mathbf{Б_C}$)

При цьому для кожного масиву $Б_{C_b}$ функції **mean**($Б_{C_b}$) і **stdev**($Б_{C_b}$) повертають, відповідно, середнє значення і стандартне відхилення його елементів.

$$Б_G := \left| \begin{array}{l} \textbf{for}\ \ b \in 1..N_S \\ \quad Б_{G_b} \leftarrow \dfrac{1024}{N} \cdot \dfrac{Б_{C_b} - \textbf{mean}(Б_{C_b})}{\textbf{stdev}(Б_{C_b})} \\ Б_G \end{array} \right. \qquad \text{(M.73)}$$

Формування прогресії (5.38) здійснюється модулем (M.74). Верхня межа 300 змінної циклу **i** та переривання останнього у разі перевищення елементом τ_i порогу 250 обрано нами умовно, виходячи з того, що в результаті трансформації (M.73) навіть найбільші значення НЧ-коефіцієнтів не перевищуватимуть межі 200...250, а для проходження усіх 300 відліків значення параметра α повинне бути меншим за 0.01, що на практиці не використовується через низьку стійкість до атак одержаної за цих умов стеганосистеми. Приклад прогресії τ наведений на рис. 5.57.

$$\tau := \left| \begin{array}{l} \textbf{for}\ \ i \in 1..300 \\ \quad \left| \begin{array}{l} \tau_i \leftarrow \left(\dfrac{1+\alpha}{1-\alpha}\right)^{i-1} \\ \textbf{break}\ \ \textbf{if}\ \ \tau_i > 250 \end{array} \right. \\ \tau \end{array} \right. \qquad \text{(M.74)}$$

$\tau =$

1	1	6	2.727	11	7.439	16	20.289	21	55.335	26	150.923
2	1.222	7	3.334	12	9.092	17	24.797	22	67.632	27	184.461
3	1.494	8	4.074	13	11.112	18	30.308	23	82.662	28	225.452
4	1.826	9	4.980	14	13.582	19	37.043	24	101.031	29	275.553
5	2.232	10	6.086	15	16.600	20	45.274	25	123.482	30	

Рис. 5.57. Прогресія τ при параметрі $\alpha = 0.1$

Індексна функція визначається програмним модулем (M.75). У пропонованому нами варіанті останній коректно опрацьовує значення аргументу $0 \le t < 1$, повертаючи заразом й індекс верхньої межі інтервалу $[\tau_i, \tau_{i+1})$ — для зручності у подальших обчисленнях.

Приклади індексних функцій за різних значень параметра α було наведено на рис. 5.53.

$$\mathbf{ind}(t) := \left| \begin{array}{l} \textbf{for } i \in 1..\,\mathbf{rows}(\tau) - 1 \\ \quad \left| \begin{array}{l} \mathbf{ind} \leftarrow 1,\ i \leftarrow 0,\ \textbf{break if } 0 \le t < \tau_1 \\ \mathbf{ind} \leftarrow (-1)^i,\ \textbf{break if } \tau_i \le t < \tau_{i+1} \end{array} \right. \\ \begin{pmatrix} \mathbf{ind} \\ i+1 \end{pmatrix} \end{array} \right. \qquad \text{(М.75)}$$

Крок 4

Проводимо ДКП трансформованих блоків зображення, використовуючи для цього програмний модуль (М.43) із заміною $\mathbf{C_b}$ на $\mathbf{Б_{C_b}}$.

Крок 5

Для зручності вбудовування, розгорнемо масив ЦВЗ у вектор, використовуючи програмний модуль (М.52). Очевидно, що загальна кількість елементів у складі отриманого вектора $\mathbf{W_{vec}}$ дорівнюватиме $\mathbf{N_W} := \mathbf{A}\cdot\mathbf{Z} = 1024$.

Крок 6

Обрання придатних для заповнення НЧ-коефіцієнтів ДКП з матриць Ω_b проводитимемо виходячи з того, що для елементів матриці, які знаходяться вище побічної діагоналі (НЧ-коефіцієнти), сума індексів є меншою за $\mathbf{N}+1$, а для тих, що нижче (ВЧ-коефіцієнти), — більшою.

Введемо мітки $\mathbf{L}$ і $\mathbf{H}$, що визначатимуть діапазон суми індексів елементів матриці Ω_b. Той елемент $(\Omega_b)_{\upsilon,\nu}$, сума індексів якого не виходить за встановлені межі: $\mathbf{L} < \upsilon+\nu < \mathbf{H}$, обиратиметься для вбудовування.

Програмний модуль пошуку мітки $\mathbf{H}$ при заданій мітці $\mathbf{L}$ — (М.76), — побудований на основі підрахунку кількості елементів, сума індексів яких задовольнятиме поставленій вимозі невиходу суми за межі $(\mathbf{L}, \mathbf{H})$. При цьому верхня межа $\mathbf{H}$ збільшується до того моменту, коли загальна кількість вказаних елементів не перевищить загальну кількість $\mathbf{N_W}$ елементів у складі вектора $\mathbf{W_{vec}}$. Наприклад, для $\mathbf{L} := 25$ (М.76) повертає значення $\mathbf{H} = 53$, що дозволяє використати для приховування $\mathbf{W_{vec}}$ 1026 НЧ-коефіцієнтів кожної з матриць Ω_b.

Вбудовування ЦВЗ до обраних НЧ-коефіцієнтів блоків зображення реалізує програмний модуль (М.77). Якщо індексна функція **ind** від абсолютного значення $\mathbf{d}$ НЧ-коефіцієнта відповідає $\mathbf{j}$-му елементу вектора ЦВЗ ($\mathbf{index} = \mathbf{W_{vec_j}}$), проводиться пошук елементів прогресії τ, на інтервалі між якими знаходиться поточне значення $\mathbf{d}$. У випадку,

якщо останнє є занадто близьким до однієї з границь інтервалу [**t1**, **t2**) (а це, як зазначалося вище, знижує стійкість створюваної стеганосистеми), коефіцієнтові $\omega_{v,\nu} = (\Omega_b)_{v,\nu}$ присвоюється нове значення — перший елемент вектора елементів, розподілених, наприклад, за нормальним законом при математичному сподіванні (**t1** + **t2**)/2 і стандартному відхиленні (**t2** − **t1**)/13. Зазначене стандартне відхилення дозволяє системі бути адаптованою до різних інтервалів $[\tau_i, \tau_{i+1})$, а також запобігти випадку, коли й нове значення коефіцієнта знову виявиться близьким до границі інтервалу. У свою чергу, обрання випадкового числа з визначеного інтервалу (а не, наприклад, значення, що відповідає середині інтервалу) унеможливлює утворення підозрілих скупчень однакових значень коефіцієнтів ДКП.

$$\begin{array}{l} \mathbf{H} := \left| \begin{array}{l} \mathbf{H} \leftarrow \mathbf{L} \\ \Sigma_{\mathbf{bit}} \leftarrow \sum_{\mathbf{i}=\mathbf{L}-1}^{\mathbf{H}-3} (\mathbf{i}+1) \\ \textbf{while}\ \ \Sigma_{\mathbf{bit}} < \mathbf{N_W} \\ \quad \left| \begin{array}{l} \mathbf{H} \leftarrow \mathbf{H}+1 \\ \Sigma_{\mathbf{bit}} \leftarrow \sum_{\mathbf{i}=\mathbf{L}-1}^{\mathbf{H}-3} (\mathbf{i}+1) \end{array} \right. \\ \mathbf{H} \end{array} \right. \end{array} \qquad \text{(M.76)}$$

Якщо результат обчислення індексної функції $\mathbf{index} \neq \mathbf{W}_{\mathbf{vec}_j}$, коефіцієнтові $\omega_{v,\nu}$ присвоюється випадкове значення з найближчого інтервалу — $[\tau_i, \tau_{i+1})$, при $i \le 2$, або $[\tau_{i-2}, \tau_{i-1})$, при $i > 2$.

На рис. 5.58 схематично зображено масиви ДКП чотирьох блоків, на які було розбито зображення: темні шумоподібні діагональні смуги у верхніх лівих квадрантах кожного з блоків відповідають модифікованим НЧ-коефіцієнтам, білі елементи — немодифікованим.

Крок 7

Для реалізації середньочастотного алгоритму даного методу для кожного елемента ЦВЗ необхідно згенерувати послідовність рівномірно розподілених на інтервалі [0, 1] ПВЧ.

За основу генератора ПВЧ можна використати ЛРЗЗЗ, реалізований програмним модулем (М.53), що генерує дійсні числа, рівномірно розподілені на інтервалі $[1, 2^d]$. Розділивши одержаний вектор **Vrnd**(...) на

(M.77)

```
Ω^ := | for b ∈ 1..N_S
      |   ω ← Ω_b , j ← 1
      |   for υ ∈ 1..N
      |     break if j > N_W
      |     for ν ∈ 1..N
      |       if L < (υ + ν) < H
      |         d ← |ω_{υ,ν}| , # ← if(ω_{υ,ν} ≥ 0 , 1 , -1)
      |         (index; i) ← ind(d)
      |         if index = W_vec_j
      |           t1 ← if(i = 1 , 0 , τ_{i-1}) , t2 ← τ_i
      |           if |d - t1| < 0.1 ∨ |d - t2| < 0.1
      |             ω_{υ,ν} ← #·rnorm(1 , (t1+t2)/2 , (t2-t1)/13)_1
      |         if index ≠ W_vec_j
      |           t1 ← τ_i , t2 ← τ_{i+1}   if i ≤ 2
      |           t1 ← τ_{i-2} , t2 ← τ_{i-1}   if i > 2
      |           ω_{υ,ν} ← #·rnorm(1 , (t1+t2)/2 , (t2-t1)/13)_1
      |         j ← j + 1
      |       break if j > N_W
      |   Ω^_b ← ω
      | Ω^
```

значення максимального його елемента (2^d), зведемо ПВЧ у діапазон [0, 1], що й виконано у програмному модулі (М.78). Останній дозволяє сформувати для кожного з $N_W^+ = N_W$ символів ЦВЗ ПВП довжиною у 2^d елементів. Початковий стан генератора ПВЧ (змінна **seed**) для першого символу ЦВЗ обрано рівним 74. Для наступних раундів він визначатиметься значенням елемента ПВП, що має індекс 74. У ході дослідження методу було встановлено, що алгоритм розширення спектра є більш надійним, якщо відношення $|\xi_j| / N_W \geq 4$. Тому попередньо приймаємо $d := \log(4 \cdot N_W, 2) = 12$. При цьому $2^d = 4096$.

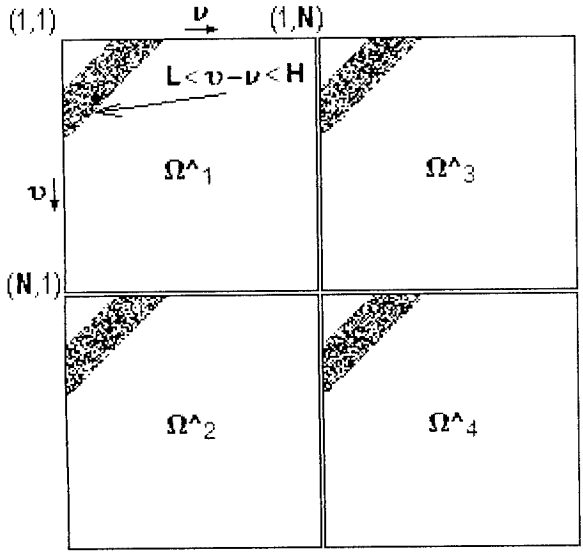


Рис. 5.58. Масиви модифікованих (зображено чорним) НЧ-коефіцієнтів ДКП блоків зображення

$$\xi := \left| \begin{array}{l} \text{seed} \leftarrow 74 \\ \text{for } j \in 1..N_W \\ \quad \left| \begin{array}{l} \xi' \leftarrow \text{Vrnd}(\text{seed}) \\ \text{seed} \leftarrow \xi'_{74} \\ \xi_j \leftarrow \dfrac{\xi'}{\max(\xi')} \end{array} \right. \\ \xi \end{array} \right. \qquad \text{(M.78)}$$

Також для генерування рівномірно розподілених ПВЧ можна скористатися й вбудованою функцією *Mathcad* **runif**(**R**, **x**, **y**), що повертає вектор з **R** випадкових чисел з рівномірним розподілом на інтервалі [**a**, **b**], — програмний модуль (М.79).

$$\xi := \left| \begin{array}{l} \text{for } j \in 1..N_W^+ \\ \quad \xi_j \leftarrow \text{runif}(4 \cdot N_W^+, 0, 1) \\ \xi \end{array} \right. \qquad \text{(M.79)}$$

Кількість СЧ-коефіцієнтів ДКП блоків зображення, до яких проводитимемо вбудовування ЦВЗ, приймемо рівною $N_{\Omega\_MF} := 4080$.

Елемент ЦВЗ, що дорівнює «+1», представлятимемо випадковим цілим числом з інтервалу [1, 8], а елемент, що має значення «–1», — випадковим цілим з інтервалу [9, 16].

З урахуванням заданих вище початкових даних, проведемо виокремлення сегмента ПВЧ η_j, що представлятиме j-й символ ЦВЗ. Для цього скористаємося програмним модулем (М.80), в якому парою функцій **round**(**runif**(...)) здійснюється генерування випадкового за рівномірним законом цілого числа на одному з вищезазначених інтервалів. За допомогою функції **submatrix**(...) виконується виокремлення з вектора ξ_j сегмента довжиною N_{Ω_MF}.

```
η := | for j ∈ 1 .. N_W+                                          (M.80)
     |   if W_vec_j = 1
     |     m ← round(runif(1, 0.5, 8.499999)_1)
     |   if W_vec_j = −1
     |     m ← round(runif(1, 8.5, 16.499999)_1)
     |   η_j ← submatrix(ξ_j, m, N_Ω_MF + m − 1, 1, 1)
     | η
```

Формуємо на основі ЦВЗ N_S відображень, кожне з яких представимо у вигляді окремої суми (5.42). Дана процедура реалізована програмним модулем (М.81).

```
Spr := | for b ∈ 1 .. N_S                                          (M.81)
       |   Spr_b ← ( Σ_{i = (N_W+/N_S)·(b−1)+1}^{(N_W+/N_S)·b} η_i − (N_W+/N_S)·1/2 ) / sqrt((N_W+/N_S)·1/12)
       | Spr
```

Результатом виконання (М.81) є вектори Spr_b, елементи яких повинні мати гаусівський розподіл з нульовим математичним сподіванням і одиничним стандартним відхиленням.

У нашому випадку для N_S блоків були одержані наступні результати:

$\mathbf{mean}(\mathbf{Spr_b}) =$

-0.001
0.012
-0.011
-0.003

$\mathbf{stdev}(\mathbf{Spr_b}) =$

1.017
0.989
0.983
1.01

Крок 8

Обрання з матриць Ω_b придатних до заповнення СЧ-коефіцієнтів проводитимемо, виходячи з того, що для елементів, які формують побічну діагональ, сума індексів дорівнює **N**+1, а сама діагональ складається з **N** елементів.

Введемо мітки **L'** і **H'**, що визначатимуть діапазон сум індексів, потрапляння у який відноситиме розгляданий СЧ-коефіцієнт до розряду тих, в які й проводитиметься вбудовування векторів $\mathbf{Spr_b}$. Для випадку $N_{\Omega\_MF} = 4080$ і $N = 128$ використання програмного модуля (М.82) дає наступні результати: $L' = 111$, $H' = 147$, що робить доступними для вбудовування 4174 СЧ-коефіцієнтів кожної з матриць Ω_b.

$$\begin{pmatrix} L' \\ H' \end{pmatrix} := \left| \begin{array}{l} L' \leftarrow N,\ H' \leftarrow N+2 \\ \Sigma_{bit} \leftarrow N,\ j \leftarrow 1 \\ \textbf{while}\ \ \Sigma_{bit} < N_{\Omega\_MF} \\ \quad \left| \begin{array}{l} \Sigma_{bit} \leftarrow \Sigma_{bit} + 2\cdot(N-j) \\ j \leftarrow j+1 \\ L' \leftarrow L'-1,\ H' \leftarrow H'+1 \end{array} \right. \\ \begin{pmatrix} L' \\ H' \end{pmatrix} \end{array} \right. \qquad \text{(М.82)}$$

Вбудовування елементів векторів $\mathbf{Spr_b}$ до обраних СЧ-коефіцієнтів ДКП блоків контейнера здійснює програмний модуль (М.83). Схематичне зображення результату вбудовування (як відмінності від оригінальних матриць Ω_b) наведене на рис. 5.59.

Крок 9

До модифікованих матриць $\Omega^{\wedge\wedge}{}_b$ застосовуємо операцію зворотного ДКП (програмний модуль (М.84)) і на основі відтворених блоків формуємо загальний масив контейнера (програмний модуль (М.85)).

$$\Omega^{\wedge\wedge} := \left|\begin{array}{l}\text{for } b \in 1..N_S \\ \quad \omega \leftarrow \Omega^{\wedge}_b,\ j \leftarrow 1 \\ \quad \text{for } \upsilon \in 1..N \\ \quad\quad \text{break if } j > N_{\Omega\_MF} \\ \quad\quad \text{for } \nu \in 1..N \\ \quad\quad\quad \text{if } L' < \upsilon+\nu < H' \\ \quad\quad\quad\quad \omega_{\upsilon,\nu} \leftarrow \omega_{\upsilon,\nu} + \gamma \cdot \left|Spr_b\right|_j \\ \quad\quad\quad\quad j \leftarrow j+1 \\ \quad\quad\quad\quad \text{break if } j > N_{\Omega\_MF} \\ \quad \Omega^{\wedge\wedge}_b \leftarrow \omega \\ \Omega^{\wedge\wedge}\end{array}\right. \qquad \text{(M.83)}$$

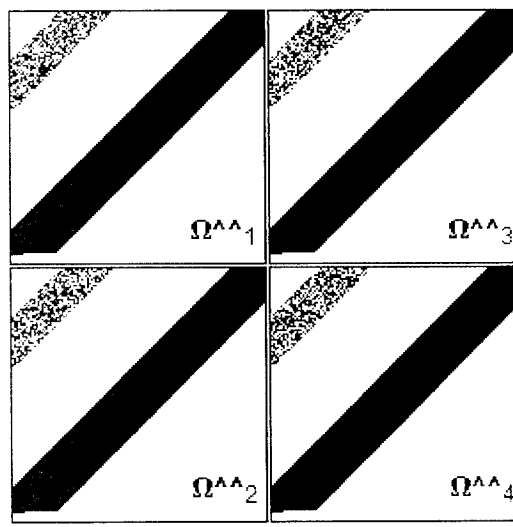


Рис. 5.59. Масиви модифікованих (зображено чорним) НЧ і СЧ-коефіцієнтів ДКП

$$Б_E := \left|\begin{array}{l}\text{for } b \in 1..N_S \\ \quad Б_{E_b} \leftarrow \dfrac{\text{idct2d}\left|\Omega^{\wedge\wedge}_b\right| \cdot N \cdot \text{stdev}\left|Б_{C_b}\right|}{1024} + \text{mean}\left|Б_{C_b}\right| \\ Б_E\end{array}\right. \qquad \text{(M.84)}$$

$$
\begin{aligned}
\mathbf{S} := \Big| \; & \textbf{for} \quad \mathbf{b} \in 1 .. \frac{\mathbf{X}}{\mathbf{N}} \\
& \quad \mathbf{S} \leftarrow \textbf{if}\left(\mathbf{b} = 1 , \mathbf{Б}_{\mathbf{E}_1} , \textbf{stack}\left(\mathbf{S} , \mathbf{Б}_{\mathbf{E}_b}\right)\right) \\
& \mathbf{S}' \leftarrow 0 \\
& \textbf{for} \quad \mathbf{b} \in \frac{\mathbf{X}}{\mathbf{N}} + 1 .. \mathbf{N}_{\mathbf{S}} \\
& \quad \Big| \; \mathbf{S}' \leftarrow \textbf{if}\left(\mathbf{S}' = 0 , \mathbf{Б}_{\mathbf{E}_b} , \textbf{stack}\left(\mathbf{S}' , \mathbf{Б}_{\mathbf{E}_b}\right)\right) \\
& \quad \Big| \; \textbf{if} \quad \textbf{mod}\left(\mathbf{b} , \frac{\mathbf{X}}{\mathbf{N}}\right) = 0 \\
& \quad \quad \Big| \; \mathbf{S} \leftarrow \textbf{augment}\left(\mathbf{S} , \mathbf{S}'\right) \\
& \quad \quad \Big| \; \mathbf{S}' \leftarrow 0 \\
& \mathbf{S} \leftarrow \textbf{scale}\left(\mathbf{S} , 0 , 255\right)
\end{aligned}
\tag{M.85}
$$

Контейнер із вбудованим ЦВЗ зображено на рис. 5.60.

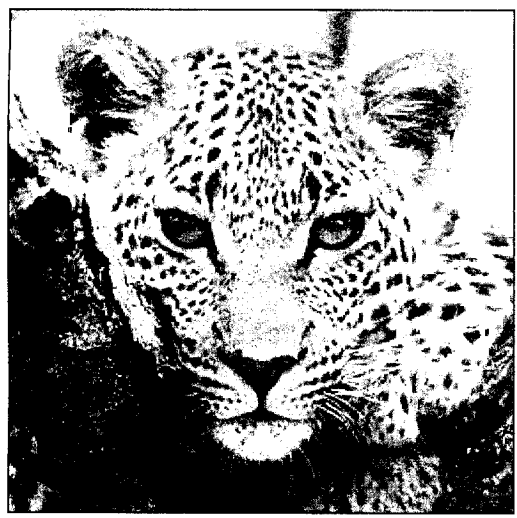


Рис. 5.60. Контейнер **S** із вбудованим ЦВЗ при $\alpha = 0.3$ і $\gamma = 1$

Крок 10

При видобуванні ЦВЗ контейнер $\mathbf{S}^* \cong \mathbf{S}$ попередньо розбивається на $\mathbf{N}^*_{\mathbf{S}} = \mathbf{N}_{\mathbf{S}} = 4$ блоки $\mathbf{Б}^*_{\mathbf{S}_b}$ розмірністю $\mathbf{N}^* \times \mathbf{N}^*$, де $\mathbf{N}^* := \mathbf{N} = 128$, — програмний модуль (М.34) з відповідною підстановкою зазначених змінних.

Кожен з отриманих блоків зображення трансформується у $\mathbf{Б}^*_{G_b}$ згідно (5.37), для чого використовується програмний модуль, подібний до (М.73), а вже для трансформованих блоків виконується пряме ДКП — за аналогічним до (М.43) програмним модулем.

Для параметра $\alpha^* = \alpha$ приймальною стороною формується прогресія τ^* і визначається індексна функція $\mathbf{ind}^*(\mathbf{t})$.

Крім того, для видобування даних з матриць ДКП Ω^*_b повинні бути заданими наступні параметри: $\mathbf{N}^*_W := \mathbf{N}_W$, $\mathbf{L}^* := \mathbf{L}$, $\mathbf{H}^* := \mathbf{H}$, $\mathbf{L}^{*'} := \mathbf{L}'$, $\mathbf{H}^{*'} := \mathbf{H}'$, $\mathbf{N}^*_{\Omega\_MF} := \mathbf{N}_{\Omega\_MF}$, $\xi^* := \xi$. Програмний модуль видобування — (М.86).

Першим циклом перебирання індексів блоків контейнера з матриць ДКП Ω^*_b обираються ті НЧ-коефіцієнти, до яких було проведено вбудовування. Ці коефіцієнти виступають аргументами індексної функції, формуючи вектор **low**. Одержаний для кожного блока результат обчислення заноситься до **b**-го елемента масиву **V**.

Наступним циклом перебирання індексів блоків з матриць ДКП обираються модифіковані СЧ-коефіцієнти, що формують вектор **mid**. Попередньо знаючи, що ЦВЗ було розподілено між усіма блоками і що елементи ЦВЗ представлені цілими числами від 1 до 16, з елемента масиву ξ, що має індекс $\mathbf{n}+(\mathbf{b}-1)\cdot\mathbf{N}^*_W/\mathbf{N}^*_S$, виокремлюються всі 16 сегментів довжиною по $\mathbf{N}^*_{\Omega\_MF}$ елементів.

Для кожного з цих сегментів обчислюється взаємна кореляція з вектором **mid** виокремлених СЧ-коефіцієнтів і проводиться пошук індексу, що відповідає найбільшому значенню кореляції. Якщо цей індекс є меншим або більшим за середину інтервалу можливих значень (8.5), робиться висновок, що вбудований елемент ЦВЗ дорівнює, відповідно, «+1» або «–1», результат чого й присвоюється векторові **v**. Останній має розмірність $(\mathbf{N}^*_W/\mathbf{N}^*_S)\times 1$ і формується окремо для кожного блока, утворюючи згодом $(\mathbf{N}^*_S+1)$-й елемент масиву **V**.

Таким чином, результуючий масив **V** складається з п'ятьох елементів, кожен з яких, у свою чергу, є вектором розмірністю $\mathbf{N}^*_W\times 1$, і, по суті, становлять собою п'ять гіпотез щодо можливого вигляду ЦВЗ (які, в ідеалі, мають бути ідентичними). Для формування єдиної гіпотези використовується «принцип більшості»: **j**-му елементу результуючого вектора $\mathbf{W}^*_{vec}$ присвоюється таке значення, що переважає серед усіх п'ятьох елементів з відповідним індексом — у пропонованому в модулі випадку з цих елементів формується вектор ω, для якого підраховується середнє значення. Якщо останнє є додатним (кількість «+1» переважає), елемент $\mathbf{W}^*_{vec_j} = 1$. Якщо від'ємним — $\mathbf{W}^*_{vec_j} = -1$. В іншому випадку $\mathbf{W}^*_{vec_j}$ псевдовипадковим чином присвоюється +1 або –1.

$$
\begin{aligned}
&W^*_{vec} := \left|\begin{array}{l}
\text{for } b \in 1..N^*_S \\
\quad j \leftarrow 1 \\
\quad \text{for } \upsilon \in 1..N^* \\
\quad\quad \text{break if } j > N^*_W \\
\quad\quad \text{for } \nu \in 1..N^* \\
\quad\quad\quad \text{if } L^* < \upsilon + \nu < H^* \\
\quad\quad\quad\quad low_j \leftarrow ind^*\left[\left|\langle \Omega^*_b \rangle_{\upsilon,\nu}\right|\right]_1 \\
\quad\quad\quad\quad j \leftarrow j + 1 \\
\quad\quad\quad \text{break if } j > N^*_W \\
\quad V_b \leftarrow low \\
\text{for } b \in 1..N^*_S \\
\quad j \leftarrow 1 \\
\quad \text{for } \upsilon \in 1..N^* \\
\quad\quad \text{break if } j > N^*_{\Omega\_MF} \\
\quad\quad \text{for } \nu \in 1..N^* \\
\quad\quad\quad \text{if } L^* < \upsilon + \nu < H^* \\
\quad\quad\quad\quad mid_j \leftarrow \langle \Omega^*_b \rangle_{\upsilon,\nu} \\
\quad\quad\quad\quad j \leftarrow j + 1 \\
\quad\quad\quad \text{break if } j > N^*_{\Omega\_MF} \\
\quad \text{for } n \in 1..\dfrac{N^*_W}{N^*_S} \\
\quad\quad \text{for } m \in 1..16 \\
\quad\quad\quad \eta \leftarrow submatrix\left[\xi^*_{\,n+\frac{N^*_W}{N^*_S}\cdot(b-1)}, m, N^*_{\Omega\_MF}+m-1, 1, 1\right] \\
\quad\quad\quad K_m \leftarrow \sum_{i=1}^{N^*_{\Omega\_MF}} \langle mid_i \cdot \eta_i \rangle \\
\quad\quad MAX \leftarrow match(max(K), K)_1 \\
\quad\quad v_n \leftarrow if(MAX < 8.5, 1, -1) \\
\quad V_{N^*_S+1} \leftarrow if\left(b = 1, v, stack\left\langle V_{N^*_S+1}, v \right\rangle\right) \\
\text{for } j \in 1..N^*_W \\
\quad \text{for } b \in 1..N^*_S + 1 \\
\quad\quad \omega_b \leftarrow \langle V_b \rangle_j \\
\quad W^*_{vec_j} \leftarrow if(mean(\omega) > 0, 1, if(mean(\omega) < 0, -1, 2 \cdot round(rnd(1)) - 1)) \\
W^*_{vec}
\end{array}\right.
\end{aligned}
\tag{M.86}
$$

Сформований вектор $\mathbf{W}^*_{vec}$ згортається у масив $\mathbf{W}^*$ з розмірністю оригінального ЦВЗ (див. програмний модуль (М.72)).

Графічне представлення видобутого ЦВЗ, вбудовування якого до контейнера було проведене за умов, що $\alpha = 0.3$ та $\gamma = 1$ (див. рис. 5.60), зображено на рис. 5.61.

Рис. 5.61. ЦВЗ, видобутий з контейнера **S**
(збільшено вдвічі)

Крок 11

Результати обчислення показників візуального спотворення для різних значень α і γ зведені до табл. 5.11 (стор. 234). Слід мати на увазі, що, порівняно з попереднім методом, контейнер було збільшено учетверо, а ЦВЗ — у стільки ж разів зменшено.

Крок 12

Результати впливу на якість видобутого ЦВЗ операції збільшення контрастності контейнера наведені на рис. 5.62.

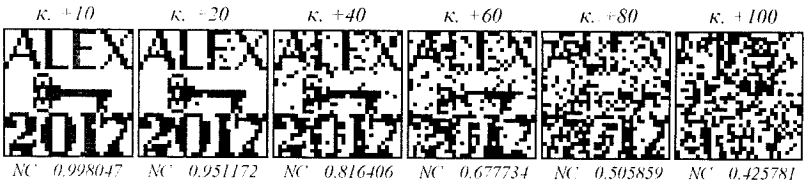


Рис. 5.62. Результат видобування ЦВЗ з контейнера **S** зі збільшеною контрастністю ($\alpha = 0.3$, $\gamma = 1$)

Крок 13

Результати видобування ЦВЗ з контейнера, обробленого Гаусівським розмиттям з різним стандартним відхиленням σ, наведені на рис. 5.63.

Рис. 5.63. Результат видобування ЦВЗ з розмитого контейнера **S** ($\alpha = 0.3$, $\gamma = 1$)

Крок 14

Результати видобування ЦВЗ зі стиснутого із втратами контейнера наведені на рис. 5.64.

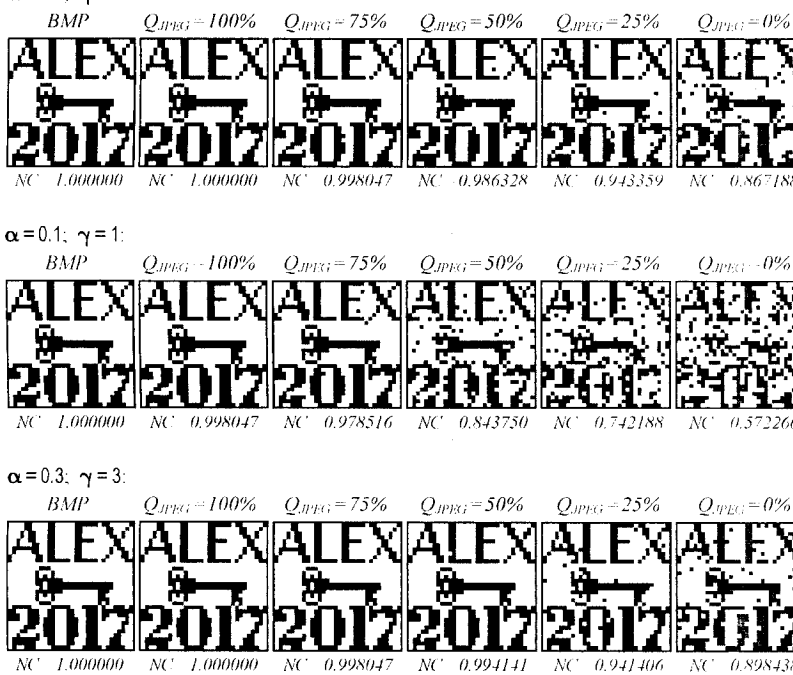


Рис. 5.64. Порівняння ЦВЗ, видобутих зі стиснутого за алгоритмом *JPEG* контейнера **S**, за різних значень α і γ

Таблиця 5.11

Показники візуального спотворення у випадку приховування даних у частотній області зображення

Назва показника спотворення	Оригінал	Методи приховування у частотній області								
		Коха і Жао: P = 1	Коха і Жао: P = 23	Бенгема та ін.: P = 1	Бенгема та ін.: P = 23	Сю і Ву: (5,29), η = 25	Сю і Ву: (5,33), η = 25	Фрідріх: α = 0.3, γ = 1	Фрідріх: α = 0.1, γ = 1	Фрідріх: α = 0.3, γ = 3
Кількість біт у повідомленні, L_M	-	176	176	176	176	4096	4096	1024	1024	1024
Кількість модифікованих пікселів	0	3624	9325	6542	10626	16045	15914	64150	63693	64687
Максимальна абсолютна різниця, *MD*	0	11	16	24	28	145	81	55	35	77
Середня абсолютна різниця, *AD*	0	0.479	1.731	1.337	2.431	17.364	12.278	12.603	9.288	20.073
Нормована середня абсолютна різниця, *NAD*	0	0.005	0.020	0.015	0.028	0.144	0.102	0.106	0.078	0.168
Середньоквадратична помилка, *MSE*	0	1.757	8.529	8.768	16.992	540.11	243.957	229.701	122.05	566.728
Нормована середньоквадратична помилка, *NMSE*	0	$1.521 \cdot 10^{-4}$	$7.388 \cdot 10^{-4}$	$7.594 \cdot 10^{-4}$	$1.472 \cdot 10^{-3}$	0.027	0.012	0.012	$6.272 \cdot 10^{-3}$	0.029
L^p-норма, $p = 2$	0	1.325	2.920	2.961	4.122	23.24	15.619	15.156	11.048	23.806
Ланласова середньоквадратична помилка, *LMSE*	0	$5.672 \cdot 10^{-3}$	0.028	0.028	0.055	1.079	0.526	0.075	0.065	0.376
Відношення «сигнал/шум», *SNR*	∞	$6.573 \cdot 10^{3}$	$1.354 \cdot 10^{3}$	$7.416 \cdot 10^{3}$	679.457	36.394	80.574	84.72	159.445	34.338
Макс. відношення «сигнал/шум», *PSNR*	∞	$3.702 \cdot 10^{4}$	$7.624 \cdot 10^{3}$	$4.848 \cdot 10^{3}$	$3.827 \cdot 10^{3}$	120.392	266.543	283.085	532.772	114.738
Якість зображення, *IF*	1	0.999848	0.999261	0.999241	0.998528	0.972523	0.987589	0.988196	0.993728	0.970878
Нормована взаємна кореляція, *NC*	1	1.000041	1.000188	0.999766	0.999659	0.988000	0.996306	0.963072	0.979165	0.932024
Якість кореляції, *CQ*	131.41	135.415	131.435	131.379	131.365	160.678	162.028	157.296	159.925	152.225
Структурний зміст, *SC*	1	0.999766	0.998886	0.999708	0.999210	0.996536	0.995002	1.066158	1.036698	1.119607
Загальне сигма-відношення «сигнал/шум», *GSSNR*	∞	$2.114 \cdot 10^{5}$	$1.44 \cdot 10^{4}$	$5.006 \cdot 10^{4}$	$2.103 \cdot 10^{4}$	255.575	151.333	20.691	43.31	10.645
Сигма-відношення «сигнал/шум», *SSNR'*	∞	68.1	100.7	95.2	120.2	108	97.3	236.266	280.892	190.346
Нормоване відношення «сигма/помилка», *NSER*	256	178	100	138	80	52.943	61.891	209.709	226.656	287.089
Подібність гістограм, *HS*	0	992	1708	1146	1604	3448	3136	22154	17836	34930

5.3.4. Методи розширення спектра

Початково методи розширення спектра (РС, *SS — Spread Spectrum*) використовувалися при розробці військових систем управління і зв'язку. Під час Другої світової війни розширення спектра використовувалося в радіолокації для боротьби з навмисними завадами, а в останні роки розвиток даної технології пояснюється бажанням створити ефективні системи радіозв'язку для забезпечення високої завадостійкості при передаванні вузькосмугових сигналів каналами з шумами й ускладнення їх перехоплення.

Система зв'язку є системою з розширеним спектром у наступних випадках [65, 115]:

- Смуга частот, використовувана під час передавання поточної інформації, є значно ширшою за мінімально для цього необхідну. При цьому енергія інформаційного сигналу розширюється на всю ширину смуги частот при низькому співвідношенні сигнал/шум, у результаті чого сигнал стає важко виявити, перехопити або ж перешкодити його передаванню шляхом внесення завад. Хоча повна потужність сигналу може бути й великою, відношення сигнал/шум у довільному діапазоні частот є малим, і це робить сигнал з розширеним спектром таким, що важко визначається при радіозв'язку і, в контексті приховування інформації стеганографічними методами, таким, що важко розрізнюється людиною.
- Розширення спектра виконується за допомогою так званого розширювального (або кодового) сигналу, що не залежить від передаваної інформації. Присутність енергії корисного сигналу в усіх частотних діапазонах робить радіосигнал з розширеним спектром стійким до внесення завад, а інформацію, вбудовану до контейнера методом розширення спектра, стійкою до її усунення чи видобування. Стиснення, кадрування та інші види атак на систему зв'язку можуть усунути енергію сигналу з деяких ділянок спектра, але, оскільки остання була поширена по всьому діапазону, в інших смугах залишиться достатньо даних для відновлення інформації. У результаті, якщо не розголошувати ключ, що використовувався для генерації кодового сигналу, ймовірність видобування повідомлення неавторизованими особами суттєво знижується.
- Відновлення початкової інформації (тобто «звуження спектра») здійснюється шляхом співставлення одержаного сигналу й синхронізованої копії кодового сигналу.

У радіозв'язку застосовують три основні способи розширення спектра:

- за допомогою прямої ПВП (РСПП);
- за допомогою стрибкоподібного перестроювання частот;
- за допомогою стиснення з використанням лінійної частотної модуляції (ЛЧМ).

При розширенні спектра прямою послідовністю інформаційний сигнал модулюється функцією, що приймає псевдовипадкові значення у встановлених межах, і помножується на часову константу — частоту (швидкість) слідування елементарних посилок, формованих елементами корисного сигналу. Модулюючий псевдовипадковий сигнал містить складові на всіх частотах, отож, при достатньому їх розширенні, здатний модулювати енергію сигналу у широкому діапазоні частот. У методі розширення спектра стрибкоподібним перестроюванням частот передавач миттєво змінює одну частоту несного сигналу на іншу. Секретним ключем при цьому є псевдовипадковий закон зміни частот. При стисненні з використанням ЛЧМ сигнал модулюється функцією, частота якої змінюється у часі. Очевидно, що будь-який із зазначених методів може бути поширений на використання в просторовій області при побудові стеганографічних систем.

Розглянемо один з варіантів реалізації методу РСПП, авторами якого є Джошуа Сміт *(J. Smith)* та Баррет Комискі *(B. Comiskey)* [88]. Алгоритм модуляції полягає в наступному: кожен біт повідомлення M_i представляється деякою базисною функцією φ_i розмірності $X \times Y$, помноженою, у залежності від значення біта («1» чи «0»), на +1 або –1:

$$\mathbf{E}(x,y) = \sum_i \mathbf{M}_i \cdot \varphi_i(x,y). \qquad (5.43)$$

Модульоване повідомлення $\mathbf{E}(x,y)$, одержане при цьому, поніксельно додається до зображення-контейнера $\mathbf{C}(x,y)$, в якості якого використовується півтонове зображення розміром $X \times Y$. Результатом є стеганозображення $\mathbf{S}(x,y) = \mathbf{C}(x,y) + \mathbf{E}(x,y)$, де $x \in 1...X$, $y \in 1...Y$.

Для унеможливлення спотворення вже вбудованого біта повідомлення, базисні функції мають бути взаємно ортогональними:

$$\left\langle \varphi_i, \varphi_j \right\rangle = \sum_{x,y}^{X,Y} \varphi_i(x,y) \cdot \varphi_j(x,y) = n_\varphi \cdot G^2 \cdot \delta_{i,j}, \qquad (5.44)$$

де n_φ — кількість значущих пікселів у базисній функції; G^2 — середня потужність, що припадає на один піксель.

$$n_\varphi \cdot G^2 = \sum_{x,y} \varphi_i^2(x,y);$$

$\delta_{i,j} = \begin{cases} 1, & \text{при } i = j; \\ 0, & \text{при } i \neq j \end{cases}$ — дельта-символ Кронекера.

В ідеальному випадку усі базисні функції φ_i повинні бути некорельованими з зображенням-контейнером **C**, іншими словами — вони мають бути ортогональними до нього: $\langle \tilde{\mathbf{N}}, \varphi_i \rangle = 0, \ \forall i$. Але на практиці знайти контейнер, що був би повністю ортогональним до всіх базисних функцій φ_i, вкрай важко. У такому випадку до розгляду вводиться величина припустимої похибки $\langle \tilde{\mathbf{N}}, \varphi_i \rangle = \Delta \approx 0$, що її, звичайно ж, необхідно враховувати належним збільшенням потужності G^2.

Для ефективного приховування інформації є потрібною значна кількість базисних функцій, ортогональних до типових зображень. Зі свого боку, кодування зображень висуває геть протилежну вимогу: ідеальною вважається саме невелика кількість базисних функцій, що приблизно перекривають усю область зображення. Ці вимоги вступають у конфлікт, коли зображення, що містить приховану інформацію, зазнає стиснення із втратами: повністю відобразити базиси, використовувані для приховування, ідеальна схема стиснення не спроможна.

Базисні функції можуть бути організовані і порівняні у відповідності до таких властивостей як повна потужність, ступінь просторового розширення (або локалізації), а також ступінь просторового частотного розширення (або локалізації) [88].

У випадку РСПП модулююча функція складається з постійного коефіцієнта підсилення G (ціле число), помноженого на псевдовипадковий блок базисних функцій φ_i (масив значень ± 1). Кожна базисна функція φ_i має індивідуальне розміщення в (x,y)-масиві. Крім того, масиви φ_i є такими, що не перетинаються (тобто є заздалегідь ортогональними один до одного) і перекривають спільний (x,y)-масив без проміжків.

Також вважатимемо, що всі N_φ базисні функції мають однакову кількість n_φ значущих елементів. У такому випадку повну потужність можна записати наступним чином:

$$P \equiv \sum_{x,y}^{X,Y} \left(\sum_i G \cdot \mathbf{M}_i \cdot \varphi_i(x,y) \right)^2 = \sum_i \sum_{x,y}^{X,Y} \left(G \cdot \mathbf{M}_i \cdot \varphi_i(x,y) \right)^2 = \\ = G^2 \cdot X \cdot Y = N_\varphi \cdot n_\varphi \cdot G^2. \qquad (5.45)$$

На етапі видобування даних стеганограми немає потреби знати первинний контейнер **C**. Операція декодування полягає у відновленні прихованого повідомлення шляхом проектування одержаного стеганозображення $\mathbf{S}^*$ на усі базисні функції φ_i:

$$\sigma_i = \left\langle \mathbf{S}^*, \varphi_i \right\rangle = \mathbf{M}_i \cdot n_\varphi \cdot G^2. \qquad (5.46)$$

Значення $\mathbf{M}_i$ можуть бути легко відновлені за допомогою наступної знакової функції:

$$\mathbf{M}_i^* = sign(\sigma_i) = \begin{cases} -1, \text{ при } \sigma_i < 0; \\ 1, \text{ при } \sigma_i > 0; \quad \text{за умови, що } G^2 >> 0. \\ ? \text{ при } \sigma_i = 0, \end{cases} \qquad (5.47)$$

Якщо $\sigma_i = 0$, то приховувана інформація була втрачена. Імовірність видобування помилкового значення біта інформації зростає за малих значень середньої потужності G^2, але при цьому менше спотворюється зображення-контейнер.

Основна перевага стеганографічних методів, що базуються на ідеї розширення спектра, — порівняно висока стійкість до різного виду атак на зображення і, оскільки приховувана інформація розподілена в широкій смузі частот, її важко видалити без повного руйнування (деградації) контейнера. Спотворення стеганозображення збільшують значення Δ і, якщо $|\Delta| < |n_\varphi \cdot G^2|$, то приховане повідомлення не постраждає.

Наведемо приклад реалізації методу стеганографічного приховання шляхом розширення спектра у програмі *Mathcad*.

Крок 1

Імпортуємо зображення-контейнер:

$$\mathbf{C} := \mathbf{READBMP}(\text{"C.bmp"}),$$

що характеризується наступними розмірами:

$$\mathbf{X} := \mathbf{rows}(\mathbf{C}) = 128;$$
$$\mathbf{Y} := \mathbf{cols}(\mathbf{C}) = 128.$$

Крок 2

Формуємо масив **Φ** ортогональних базисних функцій, що повинен мати розмірність сигналу-контейнера (**X**×**Y**) і становити собою суму всіх базисних неперекривних функцій ϕ_i.

Очевидно, що для одержання ортогональних базисних функцій ϕ_i достатньо провести поділ масиву **Φ** на неперетинні сегменти, кожен з

яких помістити у масив нульових елементів розмірністю **X×Y** за тими самими координатами.

Генерування масиву **Φ** здійснюється програмним модулем (М.87). При цьому +1 і –1 чергуються у шаховому порядку. Звичайно, що при створенні більш надійної стеганосистеми слід обрати більш складний спосіб формування масиву ортогональних базисних функцій.

```
Φ := | for x ∈ 1..X                                          (М.87)
     |  for y ∈ 1..Y
     |    Φ[x,y] ← 2·mod(x+mod(y,2),2)−1
     | Φ
```

Крок 3

Нехай загальна кількість базисних функцій $N_\phi := 256$.

Розмірність значущого підмасиву окремої базисної функції (розмірність виокремлюваного сегмента) визначимо, виходячи з розмірності масиву **Φ** і загальної кількості базисних функцій N_ϕ:

$$n := \mathrm{floor}\left(\sqrt{\frac{X \cdot Y}{N_\phi}}\right) = 8.$$

Формування масиву базисних функцій **ϕ** виконаємо шляхом виокремлення відповідного базисній функції значущого підмасиву **n×n** з загального масиву **Φ** і наступного його вбудовування у відповідні позиції нульового масиву розмірністю **X×Y** за допомогою функції **putregion**(...). Для спрощення, нульовий масив формується множенням на нуль масиву **Φ**. Зазначені операції виконуються у програмному модулі (М.88).

```
ϕ := | V ← Vrnd(seed)                                        (М.88)
     | c1 ← 1, c2 ← n
     | for i ∈ 1..N_ϕ
     |   r1 ← mod[n·(i−1)+1, X]
     |   r2 ← r1 + n − 1
     |   ϕ[|V_i|] ← putregion(0·Φ, submatrix(Φ, r1, r2, c1, c2), r1, c1)
     |   if r2 = X
     |     c1 ← c1 + n
     |     c2 ← c2 + n
     | ϕ
```

Результат присвоюється псевдовипадковому елементу масиву ϕ. Для генерування ПВП індексів використовується програмний модуль (М.53) за наступних вихідних значень:

- примітивний поліном степеня $\mathbf{d} := \mathbf{log}(\mathbf{N}_\phi, 2) = 8$, наприклад, наступного виду: $\rho(x) = 1 + x^3 + x^5 + x^7 + x^{8=d}$, що, у свою чергу, зумовлює наступний вектор вагових коефіцієнтів:

$$\mu \leftarrow (0\ 0\ 1\ 0\ 1\ 0\ 1\ 1)^{\mathbf{T}};$$

- значення початкового стану регістра **seed** — довільне ціле число на проміжку $[1; \mathbf{N}_\phi)$.

Графічне відображення масиву Φ і деяких масивів базисних функцій ϕ_i наведене на рис. 5.65.

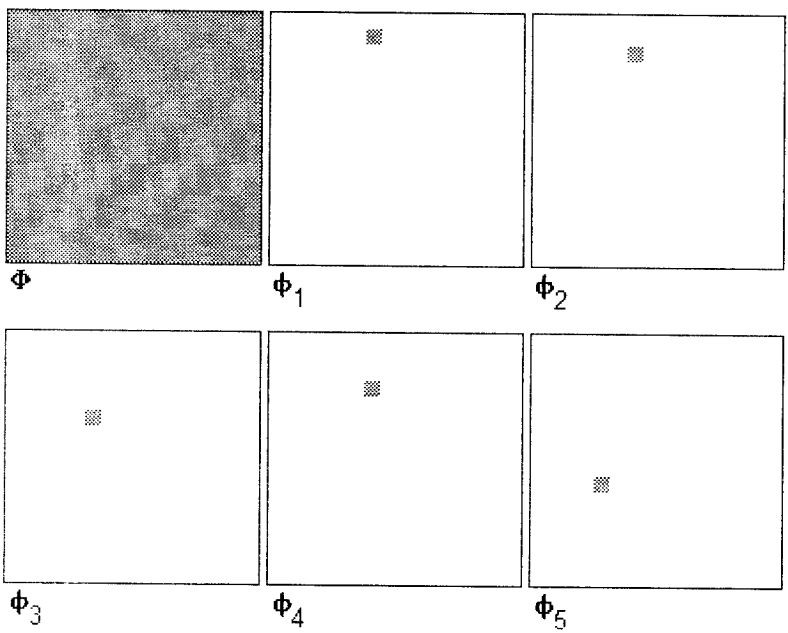


Рис. 5.65. Графічна інтерпретація масивів ортогональних базисних функцій, сформованих на основі (М.53) і (М.88) при **seed** := 23

Крок 4

Використовуючи програмний модуль (М.89), дослідимо ступінь ортогональності сигналу контейнера **C** до отриманих базисних функцій ϕ_i.

$$\Delta := \left| \begin{array}{l} \text{for } i \in 1..N_\phi \\ \quad \Delta_i \leftarrow \sum_{x=1}^{X} \sum_{y=1}^{Y} \left[\left| \phi_i \right|_{x,y} \cdot C_{x,y} \right] \\ \Delta \end{array} \right. \qquad \text{(M.89)}$$

Результат обчислення (M.89) зображено на рис. 5.66. При обраному контейнері **C** та алгоритмі формування масиву **Φ** максимальне абсолютне значення похибки ортогональності **Δ** становить 242 одиниці.

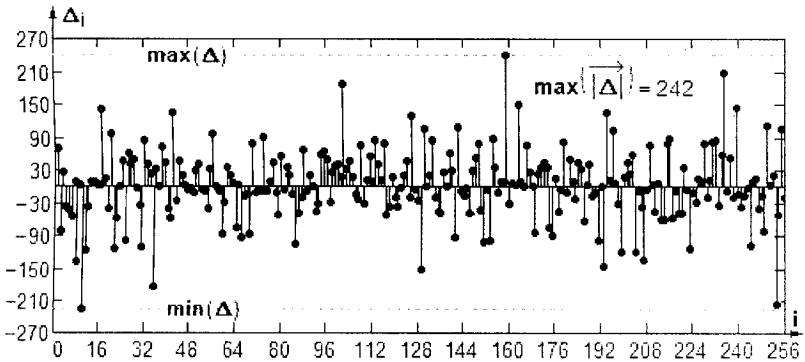


Рис. 5.66. Графік оцінювання ступеня ортогональності між масивами контейнера **C** і базисними функціями ϕ_i

Крок 5

Нехай приховуване повідомлення має наступний зміст:

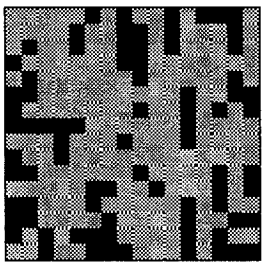
M := "© Alex Puzyrenko, 2017".

Двійкова довжина повідомлення: $L_M := 8 \cdot \mathbf{strlen}(\mathbf{M}) = 176$ бітів, що не перевищує загальну кількість наявних у нас базисних функцій $N_\phi = 256$.

Використовуючи програмний модуль (M.90), проведемо модуляцію двійкової версії повідомлення **M** базисними функціями (див. формулу (5.43)).

Результат модуляції наведено на рис. 5.67. Наявність на рисунку суцільно чорних сегментів **n**×**n** говорить про те, що деякі базисні функції не брали участі в модуляції через відсутність у повідомленні бітів з відповідними їм індексами (у нашому випадку кількість таких «невикористаних» базисних функцій: $N_\phi - L_M = 80$).

$$
\begin{aligned}
E := \;& M_{vec} \leftarrow \mathrm{str2vec}(M) \\
& \text{for } j \in 1..\mathrm{strlen}(M) \\
& \quad M_{vec\_bin} \leftarrow \mathrm{if}\left(j = 1, \mathrm{D2B}\left(M_{vec_1}, 8\right), \mathrm{stack}\left(M_{vec\_bin}, \mathrm{D2B}\left(M_{vec_j}, 8\right)\right)\right) \\
& M_{vec\_bin} \leftarrow 2 \cdot M_{vec\_bin} - 1 \\
& \text{for } x \in 1..X \\
& \quad \text{for } y \in 1..Y \\
& \quad\quad E_{x,y} \leftarrow \sum_{i=1}^{L_M} \left[\left| \phi_i \right|_{x,y} \cdot M_{vec\_bin_i} \right] \\
& E
\end{aligned}
\qquad (M.90)
$$



E

Рис. 5.67. Приклад модульованого повідомлення

Крок 6

З урахуванням максимального значення похибки Δ, визначимо достатній коефіцієнт підсилення потужності вбудовуваного до контейнера модульованого повідомлення. Для цього скористаємося спеціальним обчислювальним модулем розв'язку нерівностей (М.91), що відкривається директивою **Given**, після якої міститься логічна нерівність, у виконанні якої ми зацікавлені. Функція **Find**(K_G) повертає значення змінної K_G, попередньо заданої як $K_G := 1$, для точного розв'язку нерівності. Результат округлюється до найменшого цілого, що перевищує точний розв'язок.

$$
\begin{aligned}
& K_G := 1 \\
& n_\phi := n^2 \\
& \textbf{Given} \qquad n_\phi \cdot K_G > \max\left(\left| \overrightarrow{\Delta} \right| \right); \\
& \qquad\qquad K_G := \mathrm{ceil}\left(\mathrm{Find}\left(K_G \right) \right) = 4.
\end{aligned}
\qquad (M.91)
$$

Крок 7

З урахуванням значення коефіцієнта K_G, проводимо попереднє нормування масиву контейнера, використовуючи програмний модуль (М.92).

$$\mathbf{C_{norm}} := \left| \begin{array}{l} \textbf{for}\ \ x \in 1..X \\ \quad \textbf{for}\ \ y \in 1..Y \\ \qquad \mathbf{C_{norm}}_{x,y} \leftarrow \mathbf{round}\left[\frac{C_{x,y}}{255} \cdot \left|255 - 2 \cdot K_G\right| + K_G\right] \\ \mathbf{C_{norm}} \end{array} \right. \qquad (M.92)$$

Без проведення такої операції є досить імовірними випадки, коли в результаті вбудовування даних до контейнера значення яскравостей окремих пікселів останнього виходитимуть за межі $[0, 255]$. При цьому яскравість λ обчислюватиметься функцією запису масиву зображення до файлу як значення $\lambda' = \mathbf{mod}[\mathbf{mod}(\lambda, 256) + 256, 256]$, що розумітиме під собою відчутний зором людини стрибок значення яскравості пікселя через весь діапазон її зміни і, очевидно, є вкрай небажаним[XVIII].

Зокрема, до нормування маємо: $\mathbf{min}(\mathbf{C}) = 0$, $\mathbf{max}(\mathbf{C}) = 255$. Після виконання (М.92): $\mathbf{min}(\mathbf{C_{norm}}) = 4$, $\mathbf{max}(\mathbf{C_{norm}}) = 251$. Отож, очевидно, що навіть у випадку додавання до граничних значень яскравості пікселя контейнера $\mathbf{C_{norm}}_{x,y}$ елемента повідомлення, модульованого певною базисною функцією (що може приймати значення $\pm K_G \cdot E_{x,y}$) яскравість пікселя контейнера-результату $\mathbf{S}_{x,y}$ не вийде за припустимі межі (рис. 5.68): $\mathbf{S} := \mathbf{C_{norm}} + K_G \cdot \mathbf{E}$; $\mathbf{min}(\mathbf{S}) = 0$, $\mathbf{max}(\mathbf{S}) = 254$.

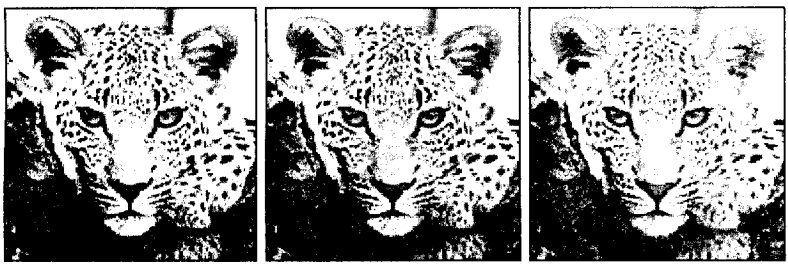


Рис. 5.68. Пустий ($\mathbf{C}$), нормований ($\mathbf{C_{norm}}$) і заповнений при $K_G = 4$ ($\mathbf{S}$) контейнери

[XVIII] Наприклад, при $\lambda = -8$ $\lambda' = 248$; при $\lambda = 259$ $\lambda' = 3$.

Крок 8

Для видобування повідомлення мають бути відомими: стеганоконтейнер $\mathbf{S}^*$, його розмірність $\mathbf{X}^*$ та $\mathbf{Y}^*$, загальна кількість базисних функцій $\mathbf{N}^*_\phi$, конфігурація (у найпростішому випадку — $\mathbf{n}^*$) і алгоритм одержання базисних функцій $\boldsymbol{\phi}^*$.

Програмний модуль видобування прихованого шляхом РСПП повідомлення згідно (5.46) і (5.47) — (М.93).

$$\mathbf{M}^* := \left| \begin{array}{l} \textbf{for}\ \ i \in 1..\mathbf{N}^*_\phi \\ \left| \begin{array}{l} \sigma \leftarrow \sum_{x=1}^{X^*} \sum_{y=1}^{Y^*} \left[\mathbf{S}^*_{x,y} \cdot \left(\boldsymbol{\phi}^*_i \right)_{x,y} \right] \\ \mathbf{M}^*_{\mathbf{vec\_bin}_i} \leftarrow \textbf{if}(\sigma > 0, 1, \textbf{if}(\sigma < 0, 0, \textbf{round}(\textbf{rnd}(1)))) \end{array} \right. \quad (*) \\ \textbf{for}\ \ j \in 1..\textbf{trunc}\left(\frac{\textbf{rows}(\mathbf{M}^*_{\mathbf{vec\_bin}})}{8} \right) \\ \left| \begin{array}{l} \mathbf{M}^*_{\mathbf{vec}_j} \leftarrow \mathbf{B2D}(\textbf{submatrix}(\mathbf{M}^*_{\mathbf{vec\_bin}}, 8 \cdot j - 7, 8 \cdot j, 1, 1)) \\ \mathbf{M}^*_{\mathbf{vec}_j} \leftarrow 32 \ \ \textbf{if}\ \ \mathbf{M}^*_{\mathbf{vec}_j} < 32 \wedge \mathbf{M}^*_{\mathbf{vec}_j} \neq 10 \wedge \mathbf{M}^*_{\mathbf{vec}_j} \neq 13 \end{array} \right. \\ \mathbf{M}^*_{\mathbf{vec}} \end{array} \right. \qquad (\text{M.93})$$

У підсумку виконання (М.93) маємо:

$$\mathbf{M}^* = \text{"© Alex Puzyrenko, 2017´ \quad Ë îV0N".}$$

Більш ефективним, на думку авторів [88], є алгоритм реалізації методу РСПП, що полягає у використанні «подвійного каналу» вбудовування: кожна базисна функція ϕ_i заздалегідь обумовленим чином ділиться на дві рівні значущі частини, що модулюються послідовностями (–1)(+1) для вбудовування «0», і (+1)(–1) для вбудовування «1». При видобуванні повідомлення з контейнера, кожен біт є результатом подвійної демодуляції — для випадків (–1)(+1) і (+1)(–1) здійснюється обчислення значень функції кореляції. Є цілком очевидним, що дійсне (за ідеального каналу зв'язку) значення біта повідомлення («0» чи «1») визначатиметься більшим значенням кореляції.

Реалізація наведеного принципу вимагає внесення змін до блоків програмних модулів вбудовування (М.90) і видобування (М.93), позначених символом "(*)". Можливий варіант заміни представлено фрагментами програмних модулів (М.94) і (М.95) відповідно.

Як на етапі вбудовування, так і під час видобування, для поділу значущих елементів базисних функцій на дві підмножини використовується значення змінної-лічильника $\mathbf{V}$, що у першому випадку (див. (М.94)) виступає в якості вектора.

(M.94)

```
⋮
E ← 0·C .  V ← 0·M_vec_bin
for x ∈ 1..X
 for y ∈ 1..Y
  for j ∈ 1..L_M
   if M_vec_bin_j = −1
    E_x,y ← E_x,y − |φ_j|_x,y   if V_j < n²/2
    E_x,y ← E_x,y + |φ_j|_x,y   if V_j ≥ n²/2
   if M_vec_bin_j = 1
    E_x,y ← E_x,y + |φ_j|_x,y   if V_j < n²/2
    E_x,y ← E_x,y − |φ_j|_x,y   if V_j ≥ n²/2
   V_j ← V_j + |(|φ_j|_x,y)|
⋮
```

(M.95)

```
⋮
σ_ ← 0 .  σ+ ← 0 .  V ← 0
for x ∈ 1..X*
 for y ∈ 1..Y*
  if V < n*²/2
   σ_ ← σ_ − S*_x,y·|φ*_i|_x,y
   σ+ ← σ+ + S*_x,y·|φ*_i|_x,y
  if V ≥ n*²/2
   σ_ ← σ_ + S*_x,y·|φ*_i|_x,y
   σ+ ← σ+ − S*_x,y·|φ*_i|_x,y
  V ← V + |(|φ*_i|_x,y)|
M*vec_bin_i ← if(σ+ > σ_, 1, if(σ+ < σ_, 0, round(rnd(1))))
⋮
```

Крок 9

Результат обчислення показників спотворення контейнера для двох розглянутих способів здійснення розширення спектра зведено до таблиці 5.12.

Таблиця 5.12

Показники візуального спотворення у випадку приховування даних методом розширення спектра

Назва показника спотворення	*Оригінал*	*РСПП № 1 (М.90), (М.93)*	*РСПП № 2 (М.94), (М.95)*
Кількість біт у повідомленні, L_M	—	176	176
Кількість модифікованих пікселів контейнера	0	15312	15311
Максимальна абсолютна різниця, *MD*	0	8	8
Середня абсолютна різниця, *AD*	0	3.401	3.402
Нормована середня абсолютна різниця, *NAD*	0	0.028	0.028
Середньоквадратична помилка, *MSE*	0	16.140	16.147
Нормована середньоквадр. помилка, *NMSE*	0	$8.211 \cdot 10^{-4}$	$8.215 \cdot 10^{-4}$
L^p-норма, при $p = 2$	0	4.017	4.018
Лапласова середньоквадр. помилка, *LMSE*	0	0.078	0.068
Відношення сигнал/шум, *SNR*	∞	$1.218 \cdot 10^3$	$1.217 \cdot 10^3$
Максимальне відношення сигнал/шум, *PSNR*	∞	$4.029 \cdot 10^3$	$4.027 \cdot 10^3$
Якість зображення, *IF*	1	0.999179	0.999179
Нормована взаємна кореляція, *NC*	1	0.993237	0.993222
Якість кореляції, *CQ*	131.41	161.529	161.527
Структурний зміст, *SC*	1	1.012869	1.012899
Загальне сигма-відношення с/ш, *GSSNR*	∞	$1.185 \cdot 10^3$	$1.171 \cdot 10^3$
Сигма-відношення сигнал/шум, *SSNR*	∞	127.3	126.6
Нормоване відношення сигнал/помилка, *NSER*	256	53.010	53.046
Подібність гістограм, *HS*	0	3136	3012

Крок 10

Результат впливу стиснення *JPEG* на можливість видобування прихованих даних для розглянутих методів розширення спектра наведено у табл. 5.13.

Таблиця 5.13

Залежність *BER* від рівня збереженої якості при *JPEG* стисненні

Якість, %		100	95	90	85	80	75	65	50	30	0
BER, %	РСПП № 1	0.000	0.000	0.000	0.000	0.000	0.000	2.273	24.432	32.955	[illegible]
	РСПП № 2	0.000	0.000	0.000	0.568	0.000	0.000	3.977	22.727	35.795	39.773

Крок 11

Результат впливу зашумлення контейнера на якість видобутих даних для розглянутих методів РСПП наведено у табл. 5.14.

Таблиця 5.14

Залежність *BER* від рівня зашумлення стеганоконтейнера

Інтенсивність:		+50	+100	+200	+400	+800
BER, %	РСПП № 1	2.273	12.500	15.909	15.909	28.977
	РСПП № 2	4.545	6.250	17.614	21.591	[illegible]

Крок 12

Результат впливу підвищення контрастності контейнера на якість видобутих даних для розглянутих методів РСПП наведено у табл. 5.15.

Таблиця 5.15

Залежність *BER* від рівня підвищення контрасту стеганоконтейнера

Контраст:		+10	+20	+50	+75	+100
BER, %	РСПП № 1	0.000	0.000	0.568	1.136	2.841
	РСПП № 2	0.568	0.568	2.273	2.841	[illegible]

Крок 13

Результат впливу кадрування контейнера на якість видобутих даних для розглянутих методів РСПП наведено у табл. 5.16.

Таблиця 5.16

Залежність *BER* від ступеня кадрування стеганоконтейнера

Товщина обрізання, *пікселів*		5	10	20
Кількість відкинутих пікселів		2460 (*15.015%*)	4720 (*28.809%*)	8640 (*52.734%*)
BER, %	РСПП № 1	1.705	11.364	17.614
	РСПП № 2	1.136	7.955	[illegible]

Крок 14

Результат впливу гаусівського розмиття контейнера на якість видобутих даних для розглянутих методів РСПП наведено у табл. 5.17.

Таблиця 5.17

Залежність *BER* від рівня гаусівського розмиття стеганоконтейнера

Станд. відхилення σ, *пікс.*		1	2	3
BER, %	РСПП № 1	0.000	9.091	38.636
	РСПП № 2	0.568	7.955	[illegible]

5.3.5. Інші методи приховування даних у статичних зображеннях

5.3.5.1. Статистичні методи

До основи статистичних методів приховування конфіденційних даних покладена модифікація певних статистичних властивостей зображення (або ж його фрагментів) з наступною перевіркою статистичних гіпотез під час видобування чи перевірки наявності зазначених даних у контейнері. Сутність статистичних методів зводиться до такої модифікації певних статистичних характеристик контейнера, за якої одержувач матиме можливість відрізнити заповнений контейнер від пустого.

Як і у вищерозглянутих методах, багаторозрядну статистичну стеганосистему можна отримати шляхом розбиття контейнера на достатню кількість неперетинних блоків (у загальному випадку їхня кількість дорівнює кількості біт L_M у приховуваному повідомленні): $b_1, \ldots, b_{L_M}$. При цьому передбачається, що до i-го блока контейнера вбудовується окремий біт повідомлення M_i. Детектування прихованого у блоці біта реалізується шляхом використання так званої перевірочної (тестової) функції. Остання дозволяє розпізнати присутність у блока ознак модифікації:

$$f(b_i) = \begin{cases} 1, \text{ якщо виявлено ознаки модифікації блока } b_i; \\ 0, \text{ якщо ознаки модифікації блока } b_i \text{ не виявлено.} \end{cases}$$

Отримання функції f є найпроблемнішою задачею при реалізації статистичного методу, оскільки її побудова здійснюється на основі теорії перевірки статистичних гіпотез.

Для роботи з даними, що мають двійковий формат, у більшості випадків проводиться оперування двома гіпотезами: основною — *«модифікації блока b_i відсутні»*, і альтернативною — *«блок b_i є результатом модифікації»*. Під час видобування прихованих даних, функцію f послідовно застосовують до всіх блоків контейнера. Якщо

статистика $q(b_i)$ розподілу елементів аналізованого блока контейнера перевищує певне порогове значення, то вважається, що до блока було вбудовано двійкову «1», у протилежному випадку — «0».

Статистичні методи доволі складно застосовувати на практиці [3, 89]. Причинами цього є, по-перше, необхідність мати вичерпну статистику $q(b_i)$ для контейнера-оригіналу, на основі якої приймаються рішення про можливу його модифікацію; по-друге, розподіл $q(b_i)$ повинен бути заздалегідь відомим на приймальному боці, що в більшості випадків є досить складною задачею.

Іоаніс Пітас *(I. Pitas)* у своїй роботі [89] пропонує використовувати статистичний метод для вбудовування до півтонового зображення **C** розмірністю $X \times Y$ цифрового підпису **W** (ЦВЗ), що являє собою псевдовипадковий двійковий шаблон розмірністю $X \times Y$, у якому кількість «одиниць» дорівнює кількості «нулів»:

$$\mathbf{W} = \left\{ w_{x,y}, x \in \{1, 2, \ldots, X\}, y \in \{1, 2, \ldots, Y\} \right\}, \tag{5.48}$$

де $w_{x,y} \in \{0, 1\}$, $\bar{w} = \frac{1}{2}$.

Оригінальне зображення представляється наступним чином:

$$\mathbf{C} = \left\{ c_{x,y}, x \in \{1, 2, \ldots, X\}, y \in \{1, 2, \ldots, Y\} \right\}, \tag{5.49}$$

де $c_{x,y} \in \{0, 1, \ldots, 255\}$ — рівень інтенсивності (яскравості) пікселя з координатами (x, y).

Множина **C** розділяється на дві підмножини рівної потужності $P = (X \times Y)/2$:

$$\mathbf{A} = \left\{ c_{x,y} \in \mathbf{C},\ w_{x,y} = 1 \right\}; \tag{5.50}$$

$$\mathbf{Z} = \left\{ c_{x,y} \in \mathbf{C},\ w_{x,y} = 0 \right\}. \tag{5.51}$$

Вбудовування ЦВЗ **W** здійснюється шляхом зміни всіх елементів підмножини **A** на величину додатного цілого коефіцієнта k:

$$\mathbf{V} = \left\{ c_{x,y} + k,\ c_{x,y} \in \mathbf{A} \right\}. \tag{5.52}$$

Зображення контейнера-результату **S** з вбудованим ЦВЗ одержується шляхом об'єднання двох множин:

$$\mathbf{S} = \mathbf{V} \cup \mathbf{Z}. \tag{5.53}$$

Незмінність зорового сприйняття зображення (непомітність вбудованих сторонніх даних) при цьому зумовлюється законом Вебера–Фехнера [75], а саме тим, що величина k, додана до яскравостей пікселів $c_{x,y} \in \mathbf{A}$ для одержання множини $\mathbf{V}$, зазвичай є достатньо малою (з урахуванням умови $k / c_{x,y} \to 0$).

І. Пітас вказує на можливість доволі точного виявлення вбудованої інформації: шляхом дослідження змін, викликаних вбудовуванням. Головна ідея при цьому полягає в експертизі відмінностей середніх значень (математичних сподівань) $\overline{v}$ і $\overline{z}$ двох виокремлених підмасивів зображення $\mathbf{V}$ і $\mathbf{Z}$ відповідно. До результатів обчислення різниці середніх значень $\overline{u} = \overline{v} - \overline{z}$ застосовується теорія перевірки гіпотези. Статистика, що лежить в основі критерію:

$$q = \overline{u} / \hat{\sigma}_{\overline{u}}, \tag{5.54}$$

де $\hat{\sigma}_{\overline{u}}^2 = \left[var(\mathbf{V}) + var(\mathbf{Z}) \right] / P$; $var(...)$ — оцінка дисперсії випадкових змінних у відповідній підмножині.

Основна й альтернативна гіпотези, відповідно, становлять собою:

H_0: ЦВЗ у зображенні відсутній ($\overline{u} = 0$);

H_1: у зображення вбудований ЦВЗ ($\overline{u} = 1$).

Виходячи з основної гіпотези, статистика q відповідає розподілові Стьюдента з нульовим математичним сподіванням і $(2 \cdot P - 2)$ ступенями свободи, що його можна з достатньою точністю апроксимувати нормальним розподілом з нульовим математичним сподіванням і одиничною дисперсією.

У випадку альтернативної гіпотези, статистика q розподілена за так званим нецентрованим розподілом Стьюдента з математичним сподіванням $k / \hat{\sigma}_{\overline{u}}$. Для великого об'єму вибірки розподіл q може бути апроксимований нормальним розподілом з одиничною дисперсією і математичним сподіванням $k / \hat{\sigma}_{\overline{u}}$.

Під час детектування наявності у контейнері ЦВЗ можлива поява помилок наступних двох типів:

- *помилка першого роду:* прийнято рішення про наявність вбудованого ЦВЗ, у той час, як останній у зображенні відсутній («хибна тривога»);
- *помилка другого роду:* наявність вбудованого ЦВЗ не підтверджена, тоді як останній у зображенні присутній («пропускання цілі»).

Якщо $t_{1-\alpha}$ є t-процентилем[XIX], що мінімізує помилки обох типів, то

$$k = \lceil 2 \cdot \hat{\sigma}_{\bar{u}} \cdot t_{1-\alpha} \rceil. \qquad (5.55)$$

де $\lceil ... \rceil$ — операція округлення до найменшого цілого, що більше або дорівнює аргументу.

Як наслідок, перед вбудовуванням ЦВЗ до контейнера виникає можливість задатися ступенем достовірності $(1-\alpha)$, з якою на стадії детектування можна зробити припущення про відсутність чи наявність вбудованого у контейнер ЦВЗ.

Отже, пропонується наступний алгоритм вбудовування ЦВЗ [89]:

1) Підраховуються значення $var(\mathbf{A})$ і $var(\mathbf{Z})$, що використовуються для визначення $\hat{\sigma}_{\bar{u}}$:

$$\hat{\sigma}_{\bar{u}} = \sqrt{[var(\mathbf{A}) + var(\mathbf{Z})]/P}.$$

2) За виразом (5.55) обчислюється значення k. Необхідно зауважити, що використане у даній формулі квантування дещо змінює ступінь достовірності, зводячи її до деякого значення $(1-\alpha')$. Крім того, автором було зроблено припущення, що $var(\mathbf{V}) = var(\mathbf{A})$, а це не є у повній мірі справедливим через відсікання, що виникають у випадку, коли результат дії $c_{x,y}+k$ виходить за межі дозволеного діапазону [0; 255].

3) Шляхом заміни підмножини **A** з **C** на підмножину **V**, створюється «підписане» ЦВЗ зображення **S** (див. формули (5.52) і (5.53)).

Алгоритм детектування ЦВЗ виглядає наступним чином:

1) Визначаються математичні сподівання $\bar{v}$ і $\bar{z}$ виокремлених підмасивів **V** і **Z**, за якими обчислюється різниця $\bar{u} = \bar{v} - \bar{z}$.

2) Визначаються оцінки дисперсії $var(\mathbf{V})$ і $var(\mathbf{Z})$, на основі яких проводиться розрахунок $\hat{\sigma}_{\bar{u}}$ (див. коментар до формули (5.54)).

3) На підставі (5.54) формується статистика q, що порівнюється з процентилями $t_{1-\alpha}$. У випадку, якщо $q < t_{1-\alpha}$, роблять висновок, що ЦВЗ у зображенні відсутній. В іншому випадку з імовірністю $(1-\alpha)$ у зображенні присутній ЦВЗ.

Розглянемо приклад одного з можливих варіантів реалізації даного методу у середовищі *Mathcad*.

[XIX] Процентиль є різновидом квантиля порядку ρ одновимірного розподілу — такого значення t_ρ випадкової величини t, для якого $P\{t < t_\rho\} = F(t_\rho) = \rho$, $(0 < \rho < 1)$. Процентилі $t_{0.01}, t_{0.02}, \ldots, t_{0.99}$ ділять область зміни ρ на 100 інтервалів, потрапляння до яких характеризується однаковою імовірністю. Більш докладно див., напр., [106, 107].

Крок 1

Нехай зображення-контейнер **C** являє собою графічний файл C.bmp (див. рис. 5.44, *а*):

$$\mathbf{C} := \mathbf{READBMP}(\text{"C.bmp"}).$$

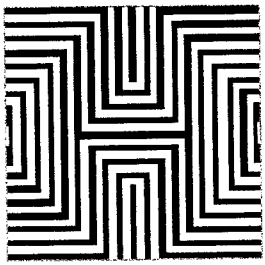
При цьому $\mathbf{X} := \mathbf{rows}(\mathbf{C}) = 128$; $\mathbf{Y} := \mathbf{cols}(\mathbf{C}) = 128$.

Чорно-біле (двійкове) зображення ЦВЗ представлено на рис. 5.69:

$$\mathbf{W} := \mathbf{READBMP}(\text{"W.bmp"}).$$

За своїм розміром ЦВЗ є ідентичним контейнерові: $\mathbf{rows}(\mathbf{W}) = 128$; $\mathbf{cols}(\mathbf{W}) = 128$.

Кількість нульових і одиничних елементів ЦВЗ є однаковим, що й підтверджує результат обчислення середнього значення множини **W**: $\mathbf{mean}(\mathbf{W}) = 0{,}5$.



W

Рис. 5.69. Приклад вбудовуваного до контейнера ЦВЗ

Крок 2

За допомогою програмного модуля (М.96) ділимо множину **C** на дві підмножини однакової потужності $\mathbf{P} := \mathbf{X} \cdot \mathbf{Y} / 2 = 8192$ у відповідності до формул (5.50) і (5.51):

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{Z} \end{pmatrix} := \left| \begin{array}{l} \textbf{for}\ \ x \in 1..\mathbf{X} \\ \quad \textbf{for}\ \ y \in 1..\mathbf{Y} \\ \quad\quad \left| \begin{array}{l} A_{x,y} \leftarrow \textbf{if}\left(W_{x,y} = 1, C_{x,y}, 0\right) \\ Z_{x,y} \leftarrow \textbf{if}\left(W_{x,y} = 0, C_{x,y}, 0\right) \end{array} \right. \\ \begin{pmatrix} \mathbf{A} \\ \mathbf{Z} \end{pmatrix} \end{array} \right. \qquad \text{(М.96)}$$

Графічна інтерпретація результату такого розбиття наведена на рис. 5.70.

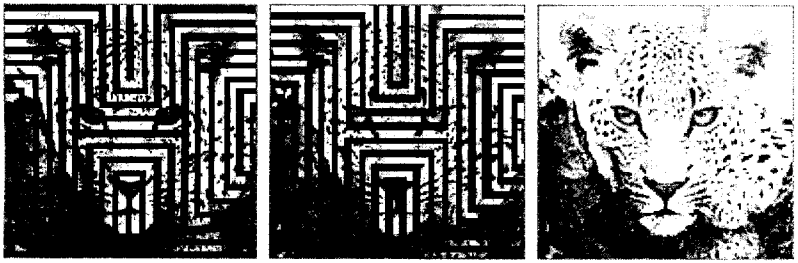


Рис. 5.70. Приклад підмножин **A** і **Z**, а також їхнього об'єднання — **A+Z**

Крок 3

Використовуючи програмні модулі (M.97) і (M.98), визначаємо, відповідно, середні значення (**mean**) і дисперсії (**var**) підмножин **A** і **Z**.

$$\begin{pmatrix} A_{mean} \\ Z_{mean} \end{pmatrix} := \left| \begin{array}{l} \Sigma A \leftarrow 0,\ \Sigma Z \leftarrow 0 \\ \text{for } x \in 1..X \\ \quad \text{for } y \in 1..Y \\ \quad\quad \Sigma A \leftarrow \Sigma A + A_{x,y} \text{ if } W_{x,y} = 1 \\ \quad\quad \Sigma Z \leftarrow \Sigma Z + Z_{x,y} \text{ otherwise} \\ \begin{pmatrix} \Sigma A \\ \Sigma Z \end{pmatrix} \cdot P^{-1} \end{array} \right. \qquad \text{(M.97)}$$

$$\begin{pmatrix} A_{var} \\ Z_{var} \end{pmatrix} := \left| \begin{array}{l} \Sigma A' \leftarrow 0,\ \Sigma Z' \leftarrow 0 \\ \text{for } x \in 1..X \\ \quad \text{for } y \in 1..Y \\ \quad\quad \Sigma A' \leftarrow \Sigma A' + \left(A_{x,y} - A_{mean}\right)^2 \text{ if } W_{x,y} = 1 \\ \quad\quad \Sigma Z' \leftarrow \Sigma Z' + \left(Z_{x,y} - Z_{mean}\right)^2 \text{ otherwise} \\ \begin{pmatrix} \Sigma A' \\ \Sigma Z' \end{pmatrix} \cdot P^{-1} \end{array} \right. \qquad \text{(M.98)}$$

У підсумку, для обраного нами контейнера маємо:

$A_{mean} = 118.4$; $Z_{mean} = 123.4$; $A_{var} = 4924.5$; $Z_{var} = 5158.4$.

За отриманими даними обчислюємо параметр $\hat{\sigma}_{\tilde{u}}$:

$$\sigma\hat{}_u := \sqrt{\frac{A_{var} + Z_{var}}{P}} = 1{,}109.$$

Крок 4

Згідно формули (5.55) встановлюємо значення **k**. Процентиль $t_{1-\alpha}$ одержимо, виходячи з того, що кількість ступенів свободи

$$d := 2 \cdot P - 2 = 16382 \approx \infty.$$

При цьому

$$t_{1-\alpha}(\alpha) := qt\left(1 - \frac{\alpha}{2}, d\right),$$

де **qt**(...) — вбудована функція *Mathcad* обчислення зворотного кумулятивного розподілу ймовірностей (квантиля) для розподілу Стьюдента.

Графік залежності $t_{1-\alpha}(\alpha)$ наведений на рис. 5.71.

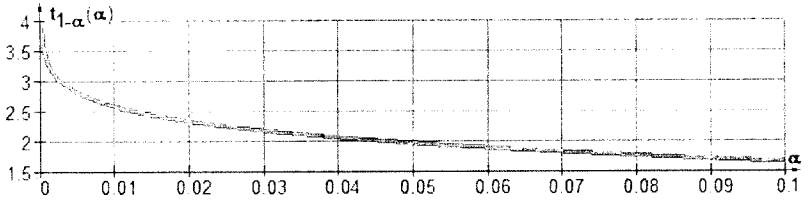


Рис. 5.71. Залежність процентиля $t_{1-\alpha}$ від ризику у припущенні помилки α

Задаючись значенням $\alpha := 0.00001$, маємо $t_{1-\alpha}(\alpha) = 5.201$, у результаті чого

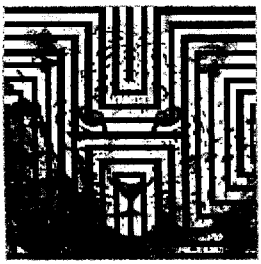
$$k(\alpha) := ceil\left(2 \cdot \sigma\hat{}_u \cdot t_{1-\alpha}(\alpha)\right) = 12.$$

Крок 5

Змінюючи елементи підмножини **A** згідно (5.52) на величину цілочислового коефіцієнта **k**, проводимо вбудовування ЦВЗ до контейнера, використовуючи для цього програмний модуль (M.99).

Використання у даному модулі вбудованої функції **clip**(...) дозволяє належним чином врахувати можливість виходу значень елементів $V_{x,y}$ за межі діапазону [0,255]. Вигляд отриманої підмножини **V**, а також «підписаного» зображення $S := V + Z$ наведені на рис. 5.72.

$$\mathbf{V} := \left| \begin{array}{l} \mathbf{k} \leftarrow \mathbf{k}(\alpha) \\ \textbf{for}\ \ x \in 1..X \\ \quad \textbf{for}\ \ y \in 1..Y \\ \quad\quad \left| \begin{array}{l} V_{x,y} \leftarrow A_{x,y} + k \quad \textbf{if}\ \ W_{x,y} = 1 \\ V_{x,y} \leftarrow 0 \quad \textbf{otherwise} \end{array} \right. \\ \textbf{clip}(\mathbf{V}, 0, 255) \end{array} \right. \qquad \text{(M.99)}$$



V

S

Рис. 5.72. Графічна інтерпретація підмножини **V** і зановленого контейнера **S**

Крок 6

Для детектування наявності ЦВЗ у контейнері мають бути відомими:

- зображення **S***, що підозрюється на наявність вбудованого ЦВЗ, і, відповідно, його розмірність **X***×**Y***;
- зображення **W***, що, як передбачається, було використане в якості ЦВЗ досліджуваного контейнера **S***.

Використовуючи програмний модуль, аналогічний модулю (M.96), ділимо множину **S*** на підмножини **V*** і **Z*** однакової потужності **P*** := **X***·**Y***/2 = 8192, керуючись відповідними значеннями елементів множини **W***.

Використовуючи програмні модулі, подібні до модулів (M.97) і (M.98), для отриманих підмножин розраховуємо середні значення (**mean**) і дисперсії (**var**):

$$\mathbf{V}^*_{\mathbf{mean}} = 130.3; \quad \mathbf{Z}^*_{\mathbf{mean}} = 123.4; \quad \mathbf{V}^*_{\mathbf{var}} = 4893.7; \quad \mathbf{Z}^*_{\mathbf{var}} = 5158.4.$$

За цими результатами знаходимо значення параметрів $\bar{u}$ і $\hat{\sigma}_{\bar{u}}$ (див. коментар до формули (5.54)):

$$\mathbf{U}_{\mathbf{mean}} := \mathbf{V}^*_{\mathbf{mean}} - \mathbf{Z}^*_{\mathbf{mean}} = 6.914 ;$$

$$\sigma^{*\wedge}{}_{U} := \sqrt{\frac{V^{*}_{var} + Z^{*}_{var}}{P^{*}}} = 1{,}108.$$

Отже, наприклад, при $\alpha^* := 0.00001$, маємо:

$$q := \frac{U_{mean}}{\sigma^{*\wedge}{}_{U}} = 6{,}242 > t^{*}_{1-\alpha}(\alpha^*) = 5{,}201.$$

З чого можна зробити висновок про те, що з імовірністю 99,999% у зображенні **S*** присутній ЦВЗ **W***.

Крок 7

Результат обчислення показників спотвореності контейнера для розглянутого методу статистичного приховування при $\alpha := 10^{-5}$ зведено до табл. 5.18.

Таблиця 5.18

Показники візуального спотворення у випадку приховування даних шляхом модифікації статистичних властивостей графічного контейнера

Назва показника спотворення	*Оригінал*	*Заповнений контейнер*
Кількість біт у ЦВЗ, L_W	—	16384
Кількість модифікованих пікселів контейнера	0	8188
Максимальна абсолютна різниця, *MD*	0	12
Середня абсолютна різниця, *AD*	0	5.940
Нормована середня абсолютна різниця, *NAD*	0	0.049
Середньоквадратична помилка, *MSE*	0	71.041
Нормована середньоквадратична помилка, *NMSE*	0	$3.614 \cdot 10^{-3}$
L^p-норма, при $p = 2$	0	8.429
Лапласова середньоквадратична помилка, *LMSE*	0	$9.997 \cdot 10^{-3}$
Відношення сигнал/шум, *SNR*	∞	276.694
Максимальне відношення сигнал/шум, *PSNR*	∞	915.314
Якість зображення, *IF*	1	0.996386
Нормована взаємна кореляція, *NC*	1	1.035373
Якість кореляції, *CQ*	131.41	168.382
Структурний зміст, *SC*	1	0.930787
Загальне сигма-відношення сигнал/шум, *GSSNR*	∞	876.058
Сигма-відношення сигнал/шум, *SSNR*	∞	133.2
Нормоване відношення сигнал/помилка, *NSER*	256	52.026
Подібність гістограм, *HS*	0	2600

5.3.5.2. *Структурні методи*

Методи, що зазнали найбільшого поширення, переважно використовують інформаційну надлишковість контейнерів на рівні пікселів (просторова область) або ж передбачають їх перетворення в частотній

області. Юрій Коростіль та Михайло Шелест запропонували метод, в якому приховування інформації здійснюється на *змістовному рівні* з використанням структурних та інформаційних параметрів зображення [3, 91]. Запропонований метод є розвитком відомої стеганографічної технології — *семаграм*. Семаграми становлять собою повідомлення, в яких шифропозначеннями є будь-які символи, окрім літери і цифр. Ці повідомлення можуть передаватися, наприклад, у малюнку, що містить крапки й тире для читання за кодом Морзе. Сутність же методу полягає у проведенні послідовних перетворень окремих фрагментів графічного зображення, що у кінцевому підсумку призводить до формування приховуваного тексту.

У складі структурних методів виокремлюють наступні етапи стеганоперетворення:

- перетворення захищуваного секретного повідомлення **M** у цифрову форму $\mathbf{D_M}$ (наприклад, за допомогою довільного криптографічного кодування, що розуміє під собою шифрування тексту з усіма відповідними атрибутами);
- перетворення послідовності чисел $\mathbf{D_M}$ на графічну структуру $\mathbf{G_M}$ (граф, піктограму тощо), що тим або іншим способом піддається формальному описові;
- перетворення графічної структури $\mathbf{G_M}$ на візуальне інформаційне середовище $\mathbf{V_M}$ (наприклад, мультимедійне чи програмне);
- використання сукупності методів і відповідних їм процедур, за допомогою яких формується сюжет з візуальних образів зі вбудованими до них прихованими повідомленнями.

Таким чином, стисло усю послідовність перетворень можна записати в такий спосіб:

$$\mathbf{M} \Rightarrow \mathbf{D_M} \Rightarrow \mathbf{G_M} \Rightarrow \mathbf{V_M} \Rightarrow \mathbf{S_M},$$

де $\mathbf{S_M}$ — опис сюжету, що складається з низки окремих графічних образів.

5.4. Приховування даних в аудіосигналах

Особливого розвитку дістали цифрові методи стеганографії в аудіосередовищі. Приховування даних у звукових (аудіо-) сигналах є особливо перспективним, оскільки слухова система людини (ССЛ) працює у надширокому динамічному діапазоні, сприймаючи більше ніж мільярд до одного у діапазоні потужності та більше ніж тисяча до одного у частотному діапазоні [14].

Разом з тим, чутливість слуху є досить гострою й до адитивного флуктуаційного (білого) шуму: відхилення у звуковому файлі можуть бути виявлені аж до однієї десятимільйонної (на 70 дБ нижче за рівень зовнішніх шумів). Та попри це, існують певні можливості для приховування інформації і в аудіосередовищі, оскільки, хоча ССЛ і має широкий динамічний діапазон, вона характеризується досить малим різницевим діапазоном. Як наслідок, гучні звуки сприяють маскуванню тихих звуків. Крім того, ССЛ не спроможна розрізнювати абсолютну фазу, розрізнюючи тільки відносну. Зрештою, існують деякі види спотворень, що викликаються оточуючим середовищем. Останні є настільки звичними для слухача, що у більшості випадків ним просто ігноруються.

Зазначені особливості слухового апарату людини дозволяють вдало використовувати аудіосередовища з метою стеганографічного захисту конфіденційної інформації. Особливий внесок у розвиток аудіостеганографії зробили Волтер Бендер *(W. Bender)*, Даніель Ґрул *(D. Gruhl)*, Норішіге Морімото *(N. Morimoto)* та Ентоні Лю *(A. Lu)*. Тому у подальшому викладенні даного підрозділу пропонується розглянути основні відомості і методи, викладені цими авторами у своїй спільній праці [14].

5.4.1. Метод заміни найменш значущих бітів аудіовідліків

Як уже було показано вище, кодування молодших розрядів є найпростішим способом ввести конфіденційні дані до складу інших структур мультимедійних даних. Використовуючи звуковий сигнал в якості контейнера, шляхом заміни НЗБ кожної точки взяття дискретного відліку, представленого двійковою кодовою комбінацією, можна зашифрувати достатньо великий об'єм інформації. Теоретично, в ідеальному відкритому каналі (без завад) ПЗ стеганоканалу складатиме 1 кб/с. на кожен кілогерц частоти дискретизації: так, в аудіопослідовності, що була оцифрована з частотою 8 кГц, бітова швидкість передавання прихованих даних складе 8 кб/с., у послідовності з частотою дискретизації 44 кГц — 44 кб/с. тощо. Платою за порівняно велику пропускну здатність каналу є відчутний на слух низькочастотний шум, відчутність якого безпосередньо залежить від вмісту (жанру) сигналу-контейнера. Наприклад, шум глядачів, що накладається на ефір спортивного змагання, у достатній мірі маскував би шум модифікованих шифруванням наймолодших бітів. Разом з цим, останній буде досить відчутним на слух у випадку звучання гри струнного квартету. Отож, є цілком очевидним, що для компенсації внесених спотворень буде доцільним використання адаптивного послаблення даних.

Головним недоліком методу заміни НЗБ, як і у випадку з графічними контейнерами, є його вкрай низька стійкість до сторонніх впливів. За виключенням, можливо, випадків, коли інформацію було вбудовано з використанням методів надлишковості, стеганограма на основі аудіофайлів може бути зруйнована звичайнісіньким шумом у реальному каналі, передискретизацією відліків тощо. Також слід зважати й на те, що стійкість до завад шляхом введення надлишковості досягається падінням швидкості передавання даних, часто — на один/два порядки. На практиці такий метод є корисним лише в замкнених, повністю цифрових середовищах, що не вимагають здійснення додаткових перетворень.

У контексті дослідження методів аудіостеганографічного кодування, зазначимо, що *Mathcad* підтримує лише файли *WAV*-формату імпульсно-кодово-модульованих (ІКМ) сигналів (*pulse-code modulated signals, PCM-signals*), який, проте, є одним з найпоширеніших на сьогодні. Перед імпортом аудіоконтейнера до документа *Mathcad* його необхідно підготувати у відповідному звуковому (музичному) редакторі.

Для запобігання можливості порівняння порушником перехопленого контейнера з наявними у нього аудіофайлами, а на цій підставі — доведення факту існування прихованого повідомлення і, можливо, видобування чи модифікації останнього, в якості контейнера рекомендується використовувати саме унікальні (створені власноруч) записи.

Підготовлений аудіоконтейнер слід помістити до поточної для формованого документа *Mathcad* директорії. У нашому випадку було попередньо створено файл ІКМ-сигналу з ім'ям "C.wav".

Крок 1

Для отримання інформації стосовно *WAV*-файлу використовується вбудована функція *Mathcad* **GETWAVİNFO**("файл"), де під аргументом "файл" розуміється текстовий рядок, що містить у собі ім'я файлу (або повний шлях і ім`я файлу), який передбачається використати в якості контейнера.

Зазначена функція повертає 4-елементний вектор з інформацією про файл, що виступив її аргументом: перший елемент вектора характеризує кількість каналів; другий — частоту дискретизації (у герцах); третій — кількість бітів, що ними кодується один аудіовідлік (і що визначає, таким чином, загальну кількість рівнів квантування); четвертий — середню кількість байтів на секунду, яку повинен обробляти аудіопрогравач, щоб відтворювати цей файл у реальному часі.

Нижче наведено приклад двох можливих варіантів використання даної функції:

$$\mathbf{GETWAVINFO}(\text{"C.wav"}) = \begin{pmatrix} 2 \\ 48000 \\ 16 \\ 192000 \end{pmatrix} \quad \text{або} \quad \begin{pmatrix} N_к \\ f_д \\ Q \\ B \end{pmatrix} \equiv \mathbf{GETWAVINFO}(\text{"C.wav"});$$

$N_к$ = 2 канал(-и);

$f_д$ = 48000 Гц;

Q = 16 біт;

B = 192000 байт/с.

Крок 2

Користуючись одержаною інформацією, можна сформувати часовий вектор, що відповідає амплітудам аудіосигналу в окремі відліки дискретизації. Ці амплітуди можуть бути зчитані за допомогою функції **READWAV**("файл"), що повертає масив, кожен стовпець якого являє собою окремий канал (зокрема, для монофонічного сигналу масив міститиме лише один стовпець, для стерео — два і т. д.), а кожен рядок масиву відповідає моментові часу, що визначається номером відліку і частотою дискретизації сигналу.

Окремий елемент аудіомасиву, в залежності від рівня квантування звуку Q, може набувати значення або від 0 до $2^8-1=255$ (при Q = 1...8), або від $-2^{16-1}=-32768$ до $2^{16-1}-1=32767$ (у випадку Q = 9...16).

Отож, нехай C := **READWAV**("C.wav"). Фрагмент імпортованого звуку (коди відліків від 250-го по 260-й у десятковому і двійковому форматі) представлено на рис. 5.73.

$i=$	$C_{i,1}=$	$C_{i,2}=$	$C_{i,1}=$	$C_{i,2}=$
250	192	53	11000000b	110101b
251	105	202	1101001b	11001010b
252	27	270	11011b	100001110b
253	-69	209	-1000101b	11010001b
254	-148	60	-10010100b	111100b
255	-180	-86	-10110100b	-1010110b
256	-133	-168	-10000101b	-10101000b
257	-1	-189	-1b	-10111101b
258	166	-198	10100110b	-11000110b
259	286	-231	100011110b	-11100111b
260	338	-251	101010010b	-11111011b

Рис. 5.73. Фрагмент імпортованого аудіофайлу у вигляді масиву квантованих амплітуд

Загальна кількість відліків, що припадає на кожен з каналів: $N_{C/к} := \mathbf{rows}(C)$, де **rows**(**C**) — функція, що повертає кількість рядків у масиві **C**. У нашому випадку: $N_{C/к} = 146994$ відл./канал.

Визначимо часові координати кожного з відліків: $n := 1..N_{C/к}$, $t_n := n/f_д$ (у секундах). Значення часових інтервалів, що відповідають наведеному вище (рис. 5.73) фрагменту відліків імпортованого звуку наведені на рис. 5.74.

i =	t_i =
250	$5.208333 \cdot 10^{-3}$
251	$5.229167 \cdot 10^{-3}$
252	$5.250000 \cdot 10^{-3}$
253	$5.270833 \cdot 10^{-3}$
254	$5.291667 \cdot 10^{-3}$
255	$5.312500 \cdot 10^{-3}$
256	$5.333333 \cdot 10^{-3}$
257	$5.354167 \cdot 10^{-3}$
258	$5.375000 \cdot 10^{-3}$
259	$5.395833 \cdot 10^{-3}$
260	$5.416667 \cdot 10^{-3}$

Рис. 5.74. Фрагмент масиву часу взяття відліків імпортованого аудіосигналу

Інтервал дискретизації

$$\Delta t := f_д^{-1} = t_n - t_{n-1}.$$

Отже, $\Delta t = 20.833 \cdot 10^{-6}$ сек., а загальна тривалість звучання аудіофайлу:

$$t_\Sigma := \mathbf{max}(t) = 3.062 \text{ сек.}$$

Очевидно, що, розділивши загальну кількість відліків $N_{C/к}$ на загальну тривалість звучання t_Σ, ми отримаємо усе те ж значення частоти дискретизації $f_д$:

$$N_{C/к}/t_\Sigma = 48000 \text{ Гц.}$$

Маючи інформацію про часові координати кожного з відліків і відповідні цим відлікам амплітуди (у квантованому вигляді), можна відтворити «осцилограму» імпортованого аудіофайлу (рис. 5.75).

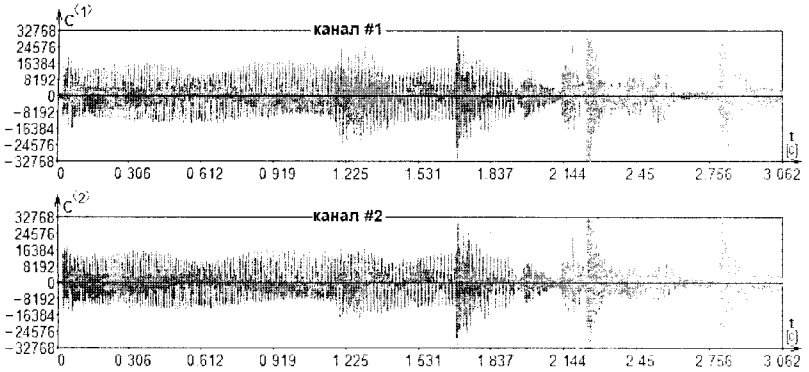


Рис. 5.75. Часові діаграми двох каналів ІКМ сигналу "C.wav"

Крок 3

В якості стеганографічного повідомлення використаємо наступне: **M** := **str2vec**("© Alex Puzyrenko, 2017"). Аналогічно до того, як це було показано вище (програмні модулі (М.1) і (М.2)), проведемо криптографічне кодування повідомлення, результатом чого буде масив **M_cod** з кодами для 22-х символів повідомлення **M**.

Крок 4

Для приховання одного символу повідомлення достатньо одного НЗБ у кодовій комбінації окремого відліку звуку. Для того, щоб при видобуванні прихованих даних з отриманого рядка символів можна було чітко визначити початок і кінець саме корисного повідомлення, введемо відповідні мітки, що тим або іншим чином вказували б на границі останнього. Приймемо, що стартова мітка μ_s визначає порядковий номер елемента контейнера, починаючи з якого в останній заноситимуться дані (див., наприклад, підпункт 5.3.2.2). Нехай $\mu_s := 107$. Про завершення корисної частини серед розпакованих символів сигналізуватиме, наприклад, мітка μ_e := "KiHeu,b" (7 символів)

Обмежуючу мітку μ_e, перетворивши її на вектор *ASCII*-кодів, введемо до тексту закодованого повідомлення:

$$\mathbf{Me} := \mathbf{stack}(\mathbf{M\_cod}, \mathbf{str2vec}(\mu_e)).$$

Таким чином, загальна кількість символів у повідомленні, що підлягає приховуванню: **rows**(**Me**) = 29 символів. Кількість потрібних для цього бітів (8 біт на один символ): 8·**rows**(**Me**) = 232 біти.

Загальна кількість НЗБ аудіоконтейнера:

$$\mathbf{rows}(\mathbf{C})\cdot\mathbf{cols}(\mathbf{C}) = 293988 \text{ бітів} >> 232 \text{ біти}.$$

Отже, аудіофайл має достатній об'єм і є придатним для використання в якості контейнера для нашого повідомлення.

Крок 5

Для подальших розрахунків також буде потрібним перетворення форматів чисел з десяткового на двійковий і назад. Для цього використаємо програмні модулі (М.3) і (М.4).

Внесення бітів повідомлення до контейнера проводитимемо зі змінним кроком, величина якого обумовлюватиметься кількістю одиниць у двійковому форматі представлення номера елемента контейнера, що модифікувався попередньо. Для визначення величини кроку використаємо програмний модуль (М.14).

Крок 6

Для зручності подальших дій, проведемо векторизацію масиву **C** (у разі, якщо кількість каналів у звуковому файлі контейнера є більшою одного) — програмний модуль (М.100), — попередньо виконавши, наприклад, перестановку елементів кожного зі стовпців у зворотному порядку, користуючись для цього вбудованою функцією **reverse**(...)).

$$\mathbf{Cv} := \left| \begin{array}{l} \textbf{for}\ \ i \in 1..N_K \\ \quad \mathbf{Cv} \leftarrow \mathbf{if}\left(i=1, \mathbf{reverse}\left(C^{(1)}\right), \mathbf{stack}\left(\mathbf{Cv}, \mathbf{reverse}\left(C^{(i)}\right)\right)\right) \\ \mathbf{Cv} \end{array} \right. \qquad \text{(М.100)}$$

На основі масиву **Cv** формуємо новий масив, що вже міститиме у собі приховане закодоване повідомлення **Me**. Для цього застосовуємо програмний модуль (М.101). При цьому кожен символ кодованого повідомлення (операція циклу **for** $\mu \in 1..$**rows**(**Me**)) переводиться у число двійкового формату (змінна **b**), кожен з восьми розрядів якого записується замість НЗБ числа, що відповідає значенню амплітуди того чи іншого відліку. Використовувані в якості контейнерів елементи масиву **Cv** проходяться не послідовно, а зі змінним кроком, величина якого у даному випадку обумовлюється кількістю одиниць у двійковому значенні номера елемента, що модифікувався попередньо. Визначальна роль при цьому відводиться функції **step**(...), що реалізується програмним модулем (М.14). Остання обчислюється при заздалегідь заданому значенні коефіцієнта **к**, що відповідає за розосередженість складових повідомлення по масиву контейнера і має враховувати як загальну кількість НЗБ, що є потрібною для вбудовування повідомлення, так і наявну кількість цих елементів у складі контейнера (у нашому випадку, наприклад, було прийнято, що **к** = 123). Стартовий елемент, нагадуємо, задається міткою μ_s.

$$\mathbf{Sv} := \left| \begin{array}{l} \mathbf{Sv} \leftarrow \mathbf{Cv} \\ z \leftarrow \mu_s \\ \textbf{for}\ \ \mu \in 1..\mathbf{rows}(\mathbf{Me}) \\ \quad b \leftarrow \mathbf{D2B}\left(\mathbf{Me}_\mu, 8\right) \\ \quad \textbf{for}\ \ i \in 1..8 \\ \quad\quad z \leftarrow z + \mathbf{step}(\mathbf{D2B}(z, \mathbf{ceil}(\log(z,2)))) \\ \quad\quad P \leftarrow \mathbf{D2B}\left(\mathbf{Cv}_z, Q\right) \\ \quad\quad * \left| \begin{array}{l} P_1 \leftarrow \mathbf{sign}\left(\mathbf{Cv}_z\right) \cdot b_i \\ \mathbf{Sv}_z \leftarrow \mathbf{B2D}(P) \end{array} \right. \\ \mathbf{Sv} \end{array} \right. \qquad \text{(М.101)}$$

Значення **z**-го елемента масиву **Cv**, до якого вестиметься запис **i**-го біта μ-го символу повідомлення **Me**, попередньо переводиться у двійковий формат (змінна **P**)). Після проведеної зміни (також враховується і знак амплітуди відліку — функція **sign**(...)) модифіковане двійкове число **P** переводиться у формат десяткового і записується у відповідну позицію вектора **Sv**, що на початку модуля був прийнятий рівним векторові **Cv**.

Крок 7

Проводимо зворотне згортання вектора **Sv** до масиву **S**, що має розмірність первинного масиву **C**. Одночасно виконується і зворотна перестановка, що має на меті повернення усіх елементів (відліків) на належні їм місця:

$$\mathbf{S} := \left| \begin{array}{l} \textbf{for } i \in 1..\textbf{cols}(\mathbf{C}) \\ \quad \mathbf{S}^{\langle i \rangle} \leftarrow \textbf{reverse}[\textbf{submatrix}[\mathbf{Sv}, (i-1)\cdot\textbf{rows}(\mathbf{C})+1, i\cdot\textbf{rows}(\mathbf{C}), 1, 1]] \\ \mathbf{S} \end{array} \right. \qquad \text{(M.102)}$$

На рис. 5.76 зображено відновлені за відповідними значеннями амплітуд відліків часові діаграми двоканального аудіоконтейнера, що містить у собі стеганограму.

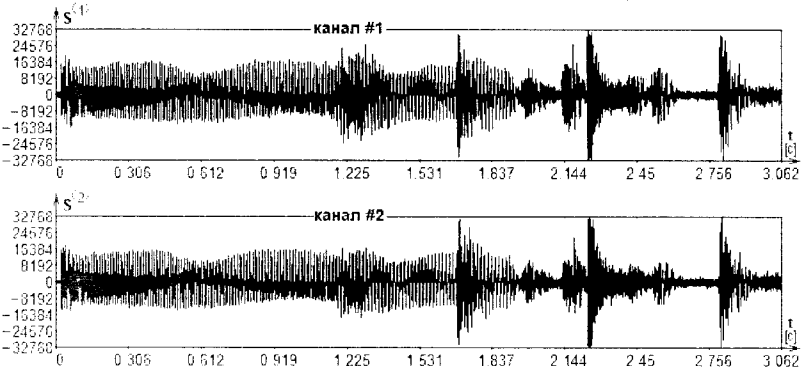


Рис. 5.76. Часові діаграми двох каналів ІКМ сигналу контейнера-результату

Порівнюючи між собою рис. 5.76 і 5.75, можна зробити висновок про їх цілковиту візуальну ідентичність, а це, у свою чергу, дає досить вагомі підстави стверджувати про ідентичність заповненого і первинного контейнерів на слух. Для того, щоб у цьому переконатися, необхідно записати масив **S** у файл, що у програмі *Mathcad* можна зробити

вбудованою функцією **WRİTEWAV**("файл", частота дискретизації, кількість рівнів квантування):

$$\textbf{WRİTEWAV}(\text{"S\_LSB.wav"}, f_д, Q) := S.$$

Крок 8

Щоб дослідити вплив на ступінь прихованості того факту, до якого з розрядів числа, що характеризує рівень сигналу певного відліку, заноситиметься конфіденційна інформація, у модулі (M.101) у поміченому символом «*» рядку замість індексу «1» слід ввести індекс, що відповідає більш старшому розряду, що модифікуватиметься.

Пересічна людина не відчуватиме на слух різниці між звучанням оригіналу і контейнера з повідомленням, якщо в якості «носіїв» використовувати не лише наймолодший, але й попередні декілька бітів (особливо, якщо використовується 16-бітове квантування і в аудіосигналі відсутні тривалі ділянки пауз або ж ділянки з тривалим постійним рівнем звучання). Отже, ще одним з можливих варіантів підсилення захисту є використання змінюваного за певним законом запису до цих декількох молодших бітів у межах окремої кодової комбінації. Або ж, жертвуючи рівнем прихованості, можна у декілька разів збільшити пропускну здатність створюваного аудіостеганоканалу.

Для наочності формованого стеганошляху, на рис. 5.77 зображено наслідок внесення даних до найстарших бітів кодових комбінацій квантованих відліків.

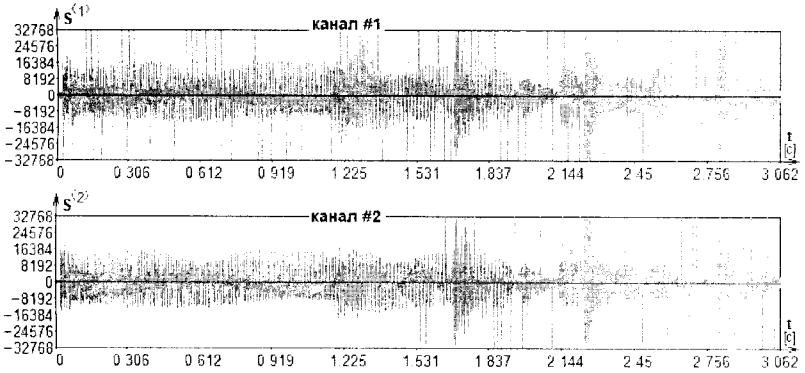


Рис. 5.77. Часові діаграми двох каналів ІКМ сигналу контейнера-результату при внесенні приховуваних даних до найстарших бітів кодових комбінацій, що характеризують рівні звучання відліків

Крок 9

Розглянемо процес видобування прихованого повідомлення. Спочатку формуємо масив значень амплітуд дискретних відліків, а також зчитуємо службову інформацію з аудіоконтейнера.

$$S^* := \mathbf{READWAV}(\text{"S\_LSB.wav"});$$

$$\begin{pmatrix} N^*_K \\ f^*_Д \\ Q^* \\ B^* \end{pmatrix} \equiv \mathbf{GETWAVINFO}(\text{"S\_LSB.wav"}); \qquad \begin{matrix} N^*_K = 2 \text{ канал(-и)}; \\ f^*_Д = 48000 \text{ Гц}; \\ Q^* = 16 \text{ біт}; \\ B^* = 192000 \text{ байт/сек}. \end{matrix}$$

Отриманий масив S^* розгортаємо у вектор Sv^*, змінюючи порядок елементів у стовпцях (для цього використовується масив (М.100) з відповідними замінами $Cv \rightarrow Sv^*$, $C \rightarrow S^*$).

Далі, використовуючи програмний модуль (М.103), проводимо безпосередньо видобування прихованого повідомлення. При цьому, оскільки у переважній більшості випадків одержувачеві заздалегідь невідомо, повідомлення якої довжини було приховано у контейнері, беруться до уваги всі елементи вектора Sv^*.

$$\begin{aligned} Mf^* := \; & z \leftarrow \mu^*_S \,, \quad err \leftarrow 0 \\ & \textbf{for} \;\; \mu \in 1 .. \, \textbf{rows}(Sv^*) \\ & \quad \textbf{for} \;\; i \in 1 .. \, 8 \\ & \quad\quad z \leftarrow z + \textbf{step}(\textbf{D2B}(z, \textbf{ceil}(\textbf{log}(z, 2)))) \\ & \quad\quad err \leftarrow 1 \,, \textbf{break} \;\; \textbf{if} \;\; z \geq \textbf{rows}(Sv^*) \\ & \quad\quad P \leftarrow \textbf{D2B}(Sv^*_z, Q^*) \\ & \quad\quad b_i \leftarrow |P_1| \\ & \quad \textbf{break} \;\; \textbf{if} \;\; err \\ & \quad Mf^*_\mu \leftarrow \textbf{B2D}(b) \\ & \quad Mf^*_\mu \leftarrow Mf^*_\mu + 32.5 \;\; \textbf{if} \;\; Mf^*_\mu < 32 \\ & Mf^* \end{aligned} \qquad \text{(М.103)}$$

Значення кожного елемента формованого при цьому вектора Mf^* представляють собою коди символів гіпотетичного повідомлення, що обчислюються у зворотному до (М.101) порядку: модуль кожного наймолодшого розряду вісімки перетворених на двійковий формат елементів вектора Sv^* (обраних з урахуванням змінного кроку і відомого значення мітки $\mu^*_S := \mu_S = 107$ формує двійкове число коду символу, формат якого потім перетворюється на десятковий. Отримане число присвоюється μ-му елементові вектора Mf^*.

Через зазначену вже неможливість оброблення *Mathcad* символів, *ASCII*-код яких має значення від 0 до 31 включно, додатково вноситься заміна відповідних значень додаванням до кожного з них коефіцієнту 32.5 (див. коментарі до програмних модулів (М.10) і (М.11)).

Змінна **err** необхідна для переривання циклу в разі виходу аргументу **z** за межі масиву $\mathbf{Sv}^*$.

Номери рядків вектора $\mathbf{Mf}^*$, елементи яких мають дробові значення, формують масив **N** (див. модуль (М.11)).

Володіючи інформацією про те, якою є кінцева мітка корисної частини повідомлення (у нашому випадку $\mu^*_e := \mu_e$ = "KiHeu,b"), виокремлюємо її з видобутого повідомлення $\mathbf{Mf}^*$, використовуючи для цього програмний модуль (М.104).

$\mathbf{M\_cod}^* :=$ (М.104)

```
e ← 0
βe ← strlen(μ*e)
Mf* ← vec2str(Mf*)
for μ ∈ 1.. strlen(Mf*)
    e ← μ  if  substr(Mf*, μ, βe) = μ*e ∧ e = 0
    break  if  e ≠ 0
M_cod* ← str2vec(substr(Mf*, 0, e))
if N
  for n ∈ 1.. cols(N)
    for i ∈ 1.. rows(N)
      break  if  N[i,n] = 0
      M_cod*[N[i,n]] ← n−1  if  0 < N[i,n] ≤ rows(M_cod*)
M_cod*
```

Вектор *ASCII*-кодів $\mathbf{Mf}^*$, попередньо перетворений функцією **vec2str**(...) на відповідний символьний рядок, послідовно проходиться у пошуку кінцевої мітки μ^*_e. Пошук здійснюється шляхом порівняння значення вказаної мітки з виокремленою частиною рядка символів, аналогічно тому, як це було реалізовано у програмному модулі (М.12). Якщо збіг відбувся, параметру **e** ставиться у відповідність значення індексу символу, що завершує змістовну частину повідомлення (μ), а цикл пошуку переривається.

Видобування частини рядка $\mathbf{Mf}^*$, обмеженої символами з індексами 0 і **e**, здійснюється за допомогою вбудованої функції **substr**(...).

Після перетворення виокремленого символьного рядка на вектор кодів *ASCII* відбувається відновлення елементів, значення яких у модулі (М.103) були примусово замінені на дробові.

Крок 10

Почергово застосовуючи модулі (М.1) і (М.13), декодуємо видобуте повідомлення. Результатом виконання модуля (М.113) є рядок символів:

$$\mathbf{vec2str}(\mathbf{M}^*) = \text{"© Alex Puzyrenko, 2017"}$$

Після декодування, повідомлення можна записати у файл:

$$\mathbf{WRITEBIN}(\text{"M\_dec.txt"}, \text{"byte"}, 0) := \mathbf{M}^*.$$

Крок 11

Результати обчислення наведених у розділі 3 (табл. 3.1) показників спотворення контейнера (адаптованих для випадку, коли останній є аудіофайлом) зведені до табл. 5.19 (стор. 302).

5.4.2. Метод фазового кодування

Чільною ідеєю методу фазового кодування є заміна фази первинного звукового сегмента на опорну фазу, характер зміни якої відтворює собою дані, які необхідно приховати. А для того, щоб між усіма сегментами зберегти різницеву фазу, фази останніх мають бути відповідним чином узгоджені.

Фазове кодування, коли воно може бути використане, є одним з найефективніших методів стеганографічного кодування за критерієм відношення сигнал/сприйманий шум. Суттєва зміна співвідношення фаз між кожними частотними складовими призводить до значного розсіювання фази. Тим не менш, до тих пір, поки модифікація фази є в достатній мірі малою (звичайно, досить малою по відношенню до конкретного спостерігача, оскільки ті зміни, що можуть здаватися незначними пересічному спостерігачеві, для фахівців зі спектрального аналізу є легко виявлюваними), може бути досягнуте таке приховання, що не буде відчутне на слух.

Процедура фазового кодування полягає в наступному:

1) звукова послідовність $\varsigma[i]$, $(1 \le i \le \dot{I})$ розбивається на серію N коротких сегментів (блоків) $\varsigma_n[i]$, $(1 \le n \le N)$ — рис. 5.78, *а*, *б*;
2) до n-го сегмента сигналу $\varsigma_n[i]$ застосовується K-точкове ДПФ, де $K = \dot{I}/N$, та створюються масиви фаз $\varphi_n(\omega_k)$ і амплітуд $A_n(\omega_k)$ для $1 \le k \le K$ (рис. 5.78, *в*);
3) запам'ятовується різниця фаз між кожними сусідніми сегментами для $1 < n \le N$ (рис. 5.78, *г*):

$$\Delta\varphi_n(\omega_k) = \varphi_n(\omega_k) - \varphi_{n-1}(\omega_k); \quad \Delta\varphi_1(\omega_k) = 0; \tag{5.56}$$

4) двійкова послідовність даних представляється як $\varphi_{data} = +\pi/2$ або $\varphi_{data} = -\pi/2$, відображуючи, відповідно, «1» або «0» (рис. 5.78, *д*):

$$\varphi'_1(\omega_k) = \varphi_{data};$$

5) з урахуванням різниці фаз відтворюється новий масив фаз для $n > 1$ (рис. 5.78, *е*):

$$\left\| \begin{array}{c} \varphi'_1(\omega_k) = \varphi_{data} \\ \varphi'_2(\omega_k) = \varphi'_1(\omega_k) + \Delta\varphi_2(\omega_k) \\ \cdots \\ \varphi'_n(\omega_k) = \varphi'_{n-1}(\omega_k) + \Delta\varphi_n(\omega_k) \\ \cdots \\ \varphi'_N(\omega_k) = \varphi'_{N-1}(\omega_k) + \Delta\varphi_N(\omega_k) \end{array} \right\|; \tag{5.57}$$

6) відновлення звукового сигналу проводиться шляхом застосування оберненого ДПФ до первинної матриці амплітуд і модифікованої матриці фаз (рис. 5.78, *ж*, *з*).

Перед процесом розшифрування має бути проведена синхронізація послідовності. Одержувачу повинні бути відомими довжина сегмента, точки ДПФ та інтервал розміщення даних стеганограми. Значення основної фази першого сегмента виявляється як «0» або «1», що представляють собою закодовану двійкову послідовність.

Оскільки фаза $\varphi'_1(\omega_k)$ є модифікованою, відповідним чином будуть змінені й абсолютні фази усіх наступних сегментів. Але відносна різниця фаз між кожними суміжними сегментами буде збережена. Са́ме до цієї відносної різниці у фазі і виявляється найбільша чутливість людського слуху.

Розкид фаз — це спотворення, викликане порушенням взаємозв'язку фаз між кожною з частотних складових. Потреба у зменшенні цього розкиду обмежує собою швидкість передавання даних під час фазового кодування. Однією з причин розкиду фаз є заміщення фази $\varphi'_1(\omega_k)$ на двійковий код приховуваного повідомлення. Для зменшення спотворень, значення модифікованої фази має бути близьким до її первинного значення. А для того щоб знизити чутливість вбудованих даних до шуму, повинна збільшуватися різниця між структурами модифікованої фази. Для цього, наприклад, значенню біта «0» відповідає $-\pi/2$, а бітові «1» — значення $+\pi/2$.

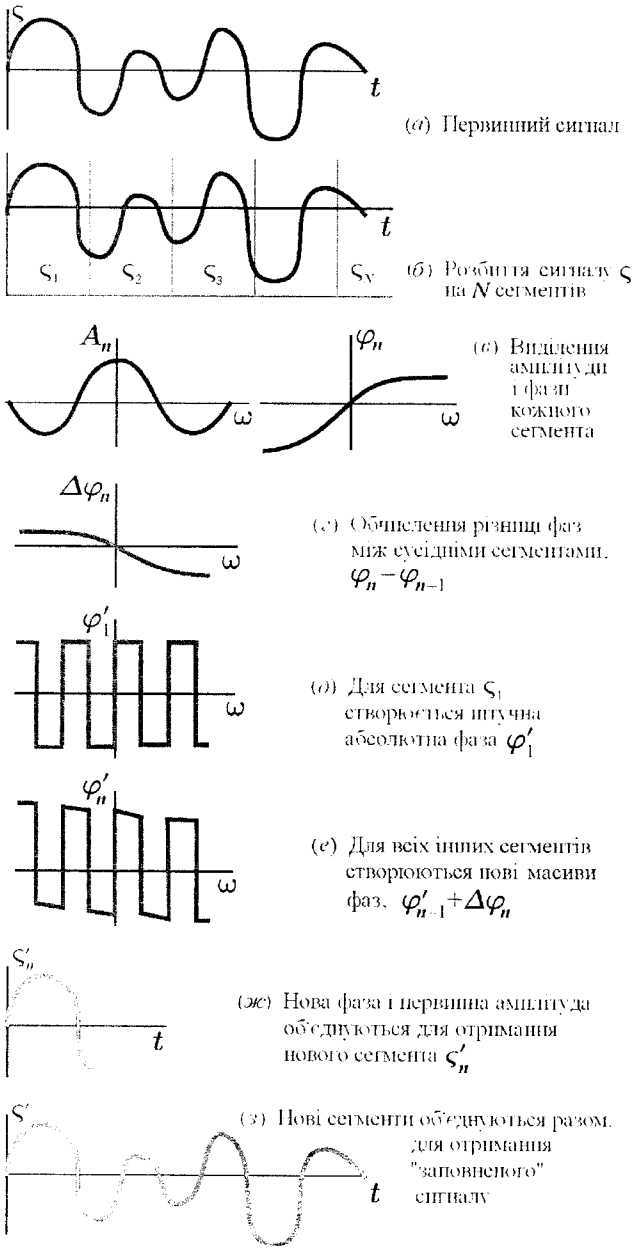


Рис. 5.78. Блок-схема фазового кодування

Ще одним джерелом спотворення є швидкість зміни модифікованої фази. Якщо спотворення буде застосоване до кожного елемента ДПФ, це з великою ймовірністю зруйнує зв'язки між фазами сусідніх частотних складових, що, у підсумку, призведе до накладання на звук фонового биття. Суттєве зниження відчутних на слух спотворювань досягається шляхом більш повільної зміни фази й узгодження переходів між змінами фаз.

На рис. 5.79 зображено різкі переходи у порівнянні зі згладженими. При цьому в обох випадках інформаційні точки відповідають однаковим значенням частот. Круті фронти фазових переходів викликають значні спотворення контейнера, тоді як, якщо зробити ці фронти більш згладженими, спотворення суттєво зменшуються.

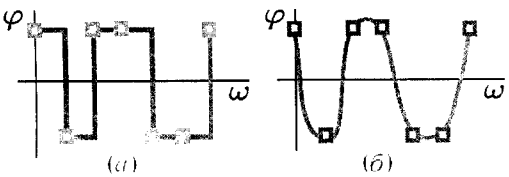


Рис. 5.79. Різкі переходи фаз (*а*) у порівнянні зі згладженими (*б*)

Слід зазначити, що така плавна зміна фазових переходів характеризується недоліком спричинення скорочення смуги пропускання, оскільки для того, щоб зробити перехід плавним, між кожною інформаційною точкою необхідно зарезервувати вдосталь місця.

Недоліком наведеної схеми фазового кодування є її низька пропускна здатність. В експериментах авторів [14] ПЗ створюваного стеганоканалу варіювалася від 8 до 32 біт/с., у залежності від звукового контексту.

Нами пропонується наступна реалізація методу.

Крок 1

Імпортуємо файл аудіоконтейнера-оригіналу до масиву квантованих амплітуд дискретних відліків у документі *Mathcad:*

$$\mathbf{C} := \mathbf{READWAV}(\text{"C.wav"}).$$

Цього разу приховувані дані вноситимемо тільки до першого каналу первинного звукового файлу: $\zeta := \mathbf{C}^{\langle 1 \rangle}$.

Кількість елементів у даному контейнері: $\mathbf{i} := \mathbf{rows}(\zeta) = 146994$; $i := 1..\mathbf{i}$. Часова діаграма даного каналу сигналу з ІКМ зображена на рис. 5.75 (*«канал #1»*).

Крок 2

Припустимо, що приховуванню підлягає повідомлення наступного змісту:

$$\mathbf{M} := \text{"© Alex Puzyrenko, 2017".}$$

Довжина повідомлення: $L_M := 8 \cdot \mathbf{strlen}(\mathbf{M}) = 176$ бітів.

Крок 3

Проведемо розбиття звукової послідовності на сегменти. Кількість сегментів **N** визначається довжиною **K** окремого сегмента. Як буде показано нижче, параметр **K** повинен бути результатом піднесення двійки до степеня, не меншого за ν, де ν — ціле число, що залежить від довжини L_M приховуваного повідомлення. При цьому біти останнього заноситимуться до масиву фаз, що одержується в результаті обчислення для сегмента швидкого перетворення Фур'є (ШПФ).

Зазначений масив має розмірність, що є вдвічі меншою за розмірність сегмента, для якого проводилося обчислення. Отже, значення параметра ν можна знайти з розв'язку наступного рівняння:

$$2^{\nu} \geq 2 \cdot L_M,$$

звідки

$$\nu := \mathbf{ceil}(\mathbf{log}(L_M, 2) + 1),$$

де $\mathbf{log}(z, 2)$ — стандартний для *Mathcad* запис обчислення логарифма за основою 2 від аргументу z; функція $\mathbf{ceil}(x)$ повертає найменше ціле, що перевищує або дорівнює аргументові x.

Результатом розв'язку при $L_M = 176$ буде $\nu = 9$. Отже, $\mathbf{K} := 2^{\nu+1}$ (степінь збільшена на одиницю для зменшення спотворюваності контейнера в результаті його заповнення), і $\mathbf{K} = 1024$; $\mathbf{k} := 1..\mathbf{K}$.

Визначимо кількість сегментів **N**, на яку необхідно поділити послідовність аудіоданих:

$$\mathbf{N} := \mathbf{ceil}(\dot{I}/\mathbf{K}).$$

Маємо: $\mathbf{N} = 144$; $\mathbf{n} := 1..\mathbf{N}$. При цьому $\dot{I}/\mathbf{K} = 143{,}549$.

Якщо ціле значення **N** перевищує собою результат $\dot{I}/\mathbf{K}$, то контейнер необхідно розширити. Наприклад, шляхом дописування у кінець вектора-сигналу нулів (програмний модуль (М.105)).

$$\begin{aligned} \mathbf{i}' &:= \dot{I} + 1 .. \mathbf{N} \cdot \mathbf{K} \\ \zeta_{\mathbf{i}'} &:= 0 \end{aligned} \qquad \text{(М.105)}$$

Таким чином, у нашому випадку нове значення $\dot{I} := \mathbf{rows}(\zeta) = 147456$.

Остаточно визначившись з основними розмірностями ключових масивів, проводимо розбиття первинної звукової послідовності ζ на сегменти, використовуючи для цього програмний модуль (М.106).

$$\varsigma := \left| \begin{array}{l} \textbf{for}\ \ n \in 1..N \\ \quad \varsigma_n \leftarrow \textbf{submatrix}[\zeta, (n-1)\cdot K+1, n\cdot K, 1, 1] \\ \varsigma \end{array} \right. \qquad \text{(М.106)}$$

Схема розбиття наведена на рис. 5.80.

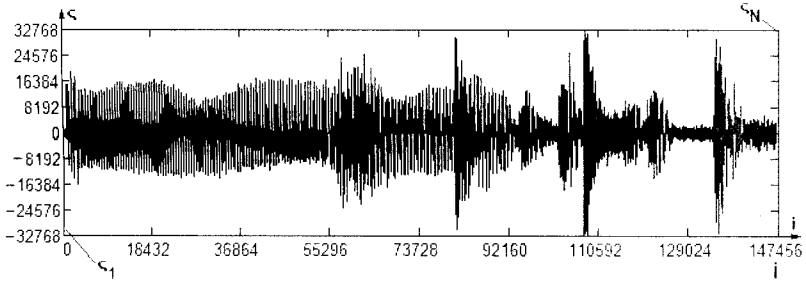


Рис. 5.80. Сигнал ζ_i ($i = 1..\dot{I}$), розбитий на N сегментів ς_n ($n = 1..N$)

Крок 4

За допомогою вбудованої функції *Mathcad* **FFT(V)** виконуємо ШПФ для даних, записаних у векторі-аргументі **V**. Останній повинен містити **K** елементів, де **K** — ціле число, значення якого було визначене на попередньому кроці. Результатом виконання функції є вектор розмірністю $[(K/2)+1]$.

У нашому випадку, в якості аргументу функції виступатимуть вектори окремих сегментів ς_n (М.107).

$$\begin{pmatrix} \sigma \\ A \\ \varphi \end{pmatrix} := \left| \begin{array}{l} \textbf{for}\ \ n \in 1..N \\ \quad \left| \begin{array}{l} \sigma_n \leftarrow \textbf{FFT}(\varsigma_n) \\ A_n \leftarrow \overrightarrow{|\sigma_n|} \\ \varphi_n \leftarrow \overrightarrow{\arg(\sigma_n)} \end{array} \right. \\ \begin{pmatrix} \sigma \\ A \\ \varphi \end{pmatrix} \end{array} \right. \qquad \text{(М.107)}$$

Запис виду $\overrightarrow{\text{operator}(V)}$ означає операцію векторизації — виконання заданої оператором дії над усіма елементами масиву **V**). У підсумку, кожен з **N** елементів масивів σ, **A** та φ являє собою підмасив з $[(K/2)+1]$ елементів, що представляють собою, відповідно, результат обчислення ШПФ, амплітуду компонента розкладу і фазовий кут для сегмента ς_n, де $n = 1..N$.

В якості прикладу, наведемо результат обчислення (лише перші 15 елементів) для сегмента № 1 (рис. 5.81).

$\varsigma_1 =$

	1
1	36
2	35
3	29
4	33
5	35
6	35
7	36
8	32
9	30
10	32
11	30
12	22
13	25
14	35
15	28
16	...

$\sigma_1 =$

	1
1	145.618
2	28.219+166.489i
3	-135.027+270.978i
4	-311.431+270.514i
5	-456.972+148.891i
6	-548.417-47.451i
7	-531.341-263.578i
8	-401.112-427.504i
9	-205.107-516.353i
10	6.113-500.409i
11	175.499-378.921i
12	250.187-216.472i
13	238.543-55.547i
14	139.057+55.924i
15	-8.371+103.81i
16	...

$A_1 =$

	1
1	145.618
2	168.864
3	302.756
4	412.513
5	480.616
6	550.466
7	593.125
8	586.217
9	555.598
10	500.447
11	417.59
12	330.838
13	244.925
14	149.881
15	104.147
16	...

$\varphi_1 =$

	1
1	0
2	1.403
3	2.033
4	2.426
5	2.827
6	-3.055
7	-2.681
8	-2.324
9	-1.949
10	-1.559
11	-1.137
12	-0.713
13	-0.229
14	0.382
15	1.651
16	...

$\overrightarrow{|A_1 \cdot \exp(j \cdot \varphi_1)|} =$

	1
1	145.618
2	28.219+166.489i
3	-135.027+270.978i
4	-311.431+270.514i
5	-456.972+148.891i
6	-548.417-47.451i
7	-531.341-263.578i
8	-401.112-427.504i
9	-205.107-516.353i
10	6.113-500.409i
11	175.499-378.921i
12	250.187-216.472i
13	238.543-55.547i
14	139.057+55.924i
15	-8.371+103.81i
16	...

Рис. 5.81. Приклад результату обчислення програмних модулів (М.106), (М.107)

Графічну інтерпретацію амплітудного і фазового масивів для сегмента № 1 представлено на рис. 5.82.

Крок 5

За допомогою програмного модуля (М.108) зберігаємо інформацію про різницю фаз між кожними сусідніми сегментами згідно (5.56).

$$\Delta\varphi := \left| \begin{array}{l} \text{for } n \in 1..N \\ \quad \Delta\varphi_n \leftarrow \text{if}(n = 1,\ 0 \cdot \varphi_1,\ \varphi_n - \varphi_{n-1}) \\ \Delta\varphi \end{array} \right. \qquad \text{(М.108)}$$

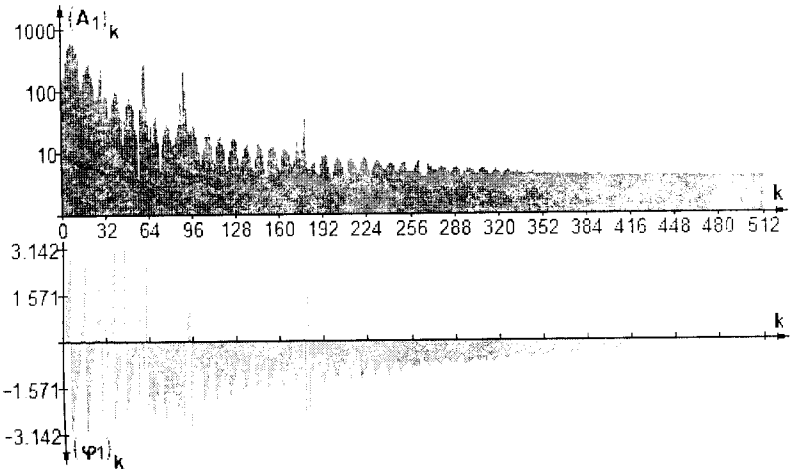


Рис. 5.82. Приклад виділеного амплітудного і фазового спектрів для сегмента ς_1

Результат обчислення різниці фаз $\Delta\varphi_2$ між сегментами № 2 і № 1 проілюстровано на рис. 5.83.

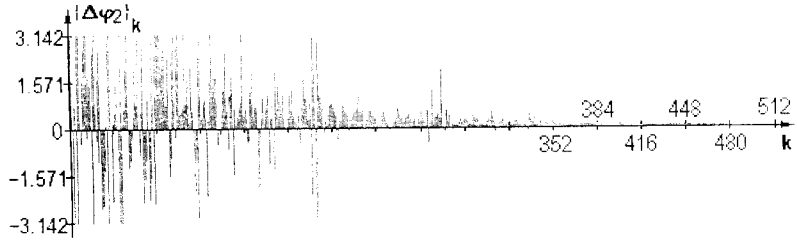


Рис. 5.83. Приклад обчислення різниці фаз між сегментами ς_2 і ς_1

Крок 6

Враховуючи різницю фаз $\Delta\varphi$, за допомогою програмного модуля (М.109) відтворюємо нову (модифіковану) матрицю фаз — φ'. При цьому бінарну послідовність, на яку попередньо перетворюємо приховувані дані **M**, вбудовуємо як значення фаз, що дорівнюють $+\pi/2$, якщо приховується біт «0» і як $-\pi/2$, якщо приховується біт «1». Результат обчислення модуля, на прикладі нових фаз сегментів ς_1 і ς_2, представлено на рис. 5.84.

$$
\begin{aligned}
\varphi' := \;& \mathbf{M} \leftarrow \mathbf{str2vec}(\mathbf{M}) \\
& \mathbf{for}\ \ \mu \in 1 .. \mathbf{rows}(\mathbf{M}) \\
& \quad \mathbf{m} \leftarrow \mathbf{if}\left(\mu = 1 ,\ \mathbf{D2B}\left(\mathbf{M}_1\right) ,\ \mathbf{stack}\left(\mathbf{m} ,\ \mathbf{D2B}\left(\mathbf{M}_\mu\right)\right)\right) \\
& \mathbf{for}\ \ k \in 1 .. \frac{K}{2} + 1 \\
& \quad \mathbf{if}\ \ k \le \mathbf{rows}(\mathbf{m}) + 1 \\
& \quad\quad \varphi'\mathbf{data}_{\frac{K}{2}+1} \leftarrow \left(\varphi_1\right)_{\frac{K}{2}+1} \quad \mathbf{if}\ \ k = 1 \vee k = \frac{K}{2} + 1 \\
& \quad\quad \varphi'\mathbf{data}_{\frac{K}{2}+2-k} \leftarrow \mathbf{if}\left(\mathbf{m}_{k-1} , -\frac{\pi}{2} , \frac{\pi}{2}\right) \quad \mathbf{if}\ \ 1 < k \le \frac{K}{2} \\
& \quad \varphi'\mathbf{data}_{\frac{K}{2}+2-k} \leftarrow \left(\varphi_1\right)_{\frac{K}{2}+2-k} \quad \mathbf{if}\ \ k > \mathbf{rows}(\mathbf{m}) + 1 \\
& \mathbf{for}\ \ n \in 1 .. N \\
& \quad \varphi'_n \leftarrow \mathbf{if}\left(n = 1 ,\ \varphi'\mathbf{data} ,\ \varphi'_{n-1} + \Delta\varphi_n\right) \\
& \varphi'
\end{aligned}
\tag{M.109}
$$

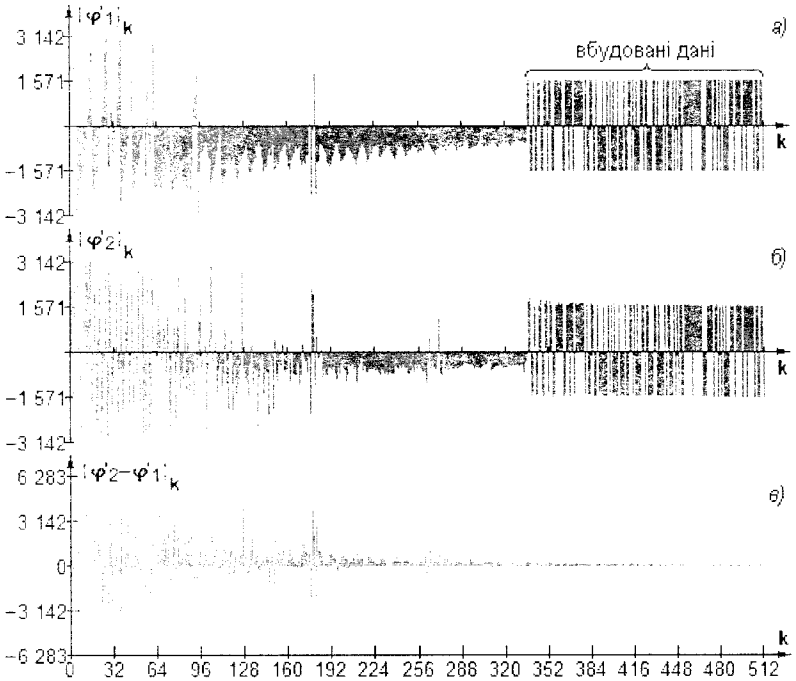


Рис. 5.84. Результати вбудовування даних до фазового спектра сегмента ς_1 (*а*), обчислення нових значень фаз спектра сегмента ς_2 (*б*) і перевірки різниці фаз між спектрами сегментів ς_2 і ς_1 (*в*)

При формуванні масиву нових фаз внесення приховуваних даних починається з ВЧ складових. Крім того, залишаються без змін перша й остання складові одержаного фазового спектра, оскільки їх модифікація призводить до значного спотворення вихідного сигналу.

Після внесення даних, що їх необхідно було приховати, масив дописується елементами з первинного фазового масиву.

Крок 7

Відновлюємо сегменти, шляхом застосування оберненого ШПФ до **n** первинних масивів амплітуд (A_n) і модифікованих масивів фаз (φ'_n) — програмний модуль (М.110). При цьому використовуємо функцію виду **iFFT(W)**, де вектор **W** має складатися з $[(K/2)+1]$ елементів. Результатом застосування даної функції є вектор **V**, кількість елементів у якому дорівнює **K**.

Відтворені модифіковані сегменти ς' об'єднуємо до спільного масиву — (М.111).

$$\varsigma' := \left| \begin{array}{l} \text{for } n \in 1..N \\ \quad \left| \begin{array}{l} \sigma \leftarrow \overrightarrow{\left| A_n \cdot \exp\left| j \cdot \varphi'_n \right| \right|} \\ \varsigma'_n \leftarrow \text{iFFT}(\sigma) \end{array} \right. \\ \varsigma' \end{array} \right. \qquad \text{(М.110)}$$

$$\zeta' := \left| \begin{array}{l} \text{for } n \in 1..N \\ \quad \zeta' \leftarrow \text{if}\left| n = 1, \varsigma'_1, \text{stack}\left| \zeta', \varsigma'_n \right| \right| \\ \zeta' \end{array} \right. \qquad \text{(М.111)}$$

Результат об'єднання зображено на рис. 5.85 (нагадаємо, що в даному випадку нами було приховано 176-бітове повідомлення).

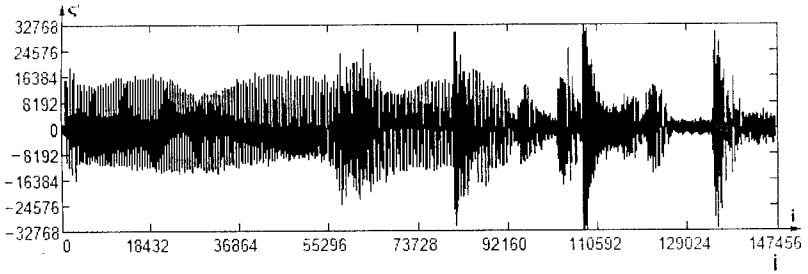


Рис. 5.85. Сигнал, відновлений за сегментами ς'_n ($n = 1..N$) при об'ємі прихованого повідомлення 176 бітів

На рис. 5.86 зображено відновлений звуковий сигнал, у випадку приховання довшого, 512-бітового повідомлення при залишенні незмінною довжини сегмента (**K** = 1024). Певна річ, в такому випадку рівень прихованості конфіденційного повідомлення є неприпустимо низьким, у чому можна було б упевнитися, просто прослухавши заповнений контейнер.

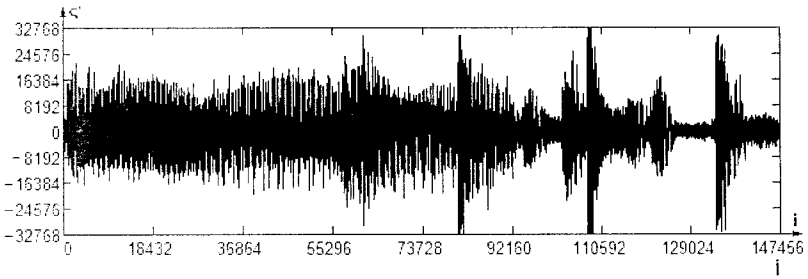


Рис. 5.86. Сигнал, відновлений за сегментами ζ'_n ($n = 1..N$) при об'ємі прихованого повідомлення 512 біт/в

Крок 8

Перед записом звукового файлу необхідно збільшити розмірність вектора, що відповідає немодифікованому каналу (у нашому випадку — другому) до розмірності модифікованого (першого). Для цього можна скористатися програмним модулем (М.112), що за необхідності дописує нулі в кінець первинного масиву.

$$\zeta'' := \left| \begin{array}{l} \zeta'' \leftarrow \left| c^{(2)} \right|,\ i \leftarrow \mathbf{rows}(\zeta'') \\ \mathbf{while}\ \ i+1 \le \mathbf{rows}(\zeta') \\ \qquad \zeta''_{i+1} \leftarrow 0,\ i \leftarrow i+1 \\ \zeta'' \end{array} \right. \qquad \text{(М.112)}$$

У результаті з'являється можливість об'єднати масиви першого і другого каналів до спільного масиву:

$$\mathbf{S} := \mathbf{augment}(\zeta', \zeta'')$$

і, зрештою, провести запис утворених даних до файлу:

$$\mathbf{WRITEWAV}(\text{"S\_Phase.wav"}, f_д, \mathbf{Q}) := \mathbf{S}.$$

Крок 9

Розглянемо алгоритм видобування прихованої інформації. На основі аудіофайлу, що являє собою заповнений стеганоконтейнер, формуємо масив квантованих амплітуд дискретних відліків:

$$S^* := \mathbf{READWAV}(\text{"S\_Phase.wav"}).$$

Одержувачеві повинні бути відомими:

- довжина кожного з сегментів (у нашому випадку — $K^* := 1024$);
- точки ШПФ, до яких вбудовувалися дані повідомлення;
- стеганошлях (канал, до якого заносилася інформація, алгоритм її занесення тощо).

Нехай відомо, що приховані дані можуть міститися у першому каналі. Отже, $\zeta^* := S^{*<1>}$. Загальна кількість елементів у даному контейнері: $l^* := \mathbf{rows}(\zeta^*) = 147456$. Кількість сегментів, на яку необхідно поділити контейнер: $N^* := l^*/K^* = 144$.

Відповідно до програмного модуля (М.106) проводимо розбиття звукової послідовності ζ^* на сегменти ς^*_n.

Для першого сегмента обчислюємо ШПФ і визначаємо масив фаз:

$$\sigma^*_1 := \mathbf{FFT}(\varsigma^*_1); \qquad \varphi^*_1 := \overrightarrow{\mathbf{arg}(\sigma^*_1)}.$$

Щоб видобути приховану інформацію з отриманого масиву фаз, використовуємо програмний модуль (М.113).

$$\begin{aligned}
M^* := \ & \mathbf{for}\ \ \mu \in 1 .. \frac{K^*}{2} \\
& \quad d \leftarrow \left|\varphi^*_1\right|_{\frac{K^*}{2}+1-\mu} \\
& \quad b_\mu \leftarrow \mathbf{if}(d < 0, 1, 0) \\
& \mathbf{for}\ \ i \in 1 .. \mathbf{trunc}\left(\frac{\mathbf{rows}(b)}{8}\right) \\
& \quad B \leftarrow \mathbf{submatrix}[b, (i-1)\cdot 8+1, i\cdot 8, 1, 1] \\
& \quad M^*_i \leftarrow \mathbf{B2D}(B) \\
& \quad M^*_i \leftarrow 32 \ \ \mathbf{if}\ \ M^*_i < 32 \\
& \mathbf{vec2str}(M^*)
\end{aligned} \qquad \text{(М.113)}$$

У даному модулі послідовно аналізуються значення фаз кожної частотної складової спектра першого сегмента на факт перевищення встановленого порогу (як у бік від'ємних, так і в бік додатних значень). В якості порогових значень не рекомендується використову-

вати точну рівність $\pm\pi/2$, оскільки під час передавання чи зберігання аудіофайл може зазнати певних змін. Крім того, операції ШПФ повертають у певній мірі округлені значення. Виходячи з цього поріг рекомендується знизити (у нашому випадку було встановлено рівень 0 рад.). Тоді при від'ємному значенні фази приймається рішення, що прихований біт дорівнює «1», у випадку ж додатного значення фази робиться висновок про приховання біту «0».

Результатом виконання (М.113) є рядок символів:[XX]

M* = "© Alex Puzyrenko, 2017ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿïþÿÿÿÿÿÿÿÿÿÿÇÿÿÿcÿã   þø ".

Крок 10

Обчислені показники звукового спотворення зведено до табл. 5.19 (стор. 302).

5.4.3. Метод розширення спектра

У типовому каналі зв'язку часто виявляється бажаним зосередити інформацію у якомога вужчому діапазоні частот, наприклад, для того щоб задовольнитися наявною смугою пропускання (прозорості) каналу і зменшити потужність сигналу (для запобігання перевищення об'єму сигналу ємності каналу зв'язку [115]). З іншого боку, метод розширення спектра переважно призначений для шифрування потоку інформації шляхом «розсіювання» кодованих даних по всьому можливому частотному спектру — це робить можливим прийом сигналу навіть за наявності на певних частотах потужних завад.

У [14] розглядається технологія розширення спектра сигналу прямою послідовністю (РСПП). Як уже зазначалося вище (див. п. 5.3.4), методи РСПП розширюють сигнал приховуваних даних (повідомлення), помножуючи його на елементарну посилку (т. зв. чип-сигнал) — ПВП максимальної довжини, модульовану відомою частотою.

Оскільки використовувані в якості контейнерів аудіосигнали мають цифровий формат, то в якості частоти елементарної посилки під час кодування можна використовувати частоту дискретизації. Як наслідок, дискретний характер сигналу усуває найскладнішу проблему, що виникає при одержанні сигналу з розширеним прямою послідовністю спектром, — коректне визначення початку і кінця складових елементарної посилки з метою фазової синхронізації. Отже, виникає можливість використання набагато вищої частоти слідування елементарних посилок, а, відтак, й

[XX] Оскільки обмежувальна мітка кінця повідомлення не використовувалася, програмний модуль досліджує усі $K^*/2$ складові спектра першого сегмента.

отримання більш високої пов'язаної з нею швидкості передавання даних. До того ж можуть застосовуватися ще й різноманітні алгоритми блокування сигналу, але в обчислювальному сенсі вони є досить складними.

У РСПП для шифрування і дешифрування інформації необхідний один і той самий ключ — псевдовипадковий шум, що в ідеальному випадку має постійну частотну характеристику в усьому діапазоні частот (так званий білий шум). Ключ застосовується до приховуваної інформації і трансформує її послідовність у послідовність з розширеним спектром.

Власне по відношенню до аудіосигналів метод РСПП полягає в наступному. Сигнал даних (приховуване повідомлення) помножується на сигнал-носій і псевдовипадкову шумову послідовність, що характеризується рівномірно широкою спектральною потужністю. У результаті спектр даних розширюється на всю доступну смугу, після чого послідовність розширених даних послаблюється і додається до первинного файлу (контейнера) як адитивний випадковий шум (рис. 5.87).

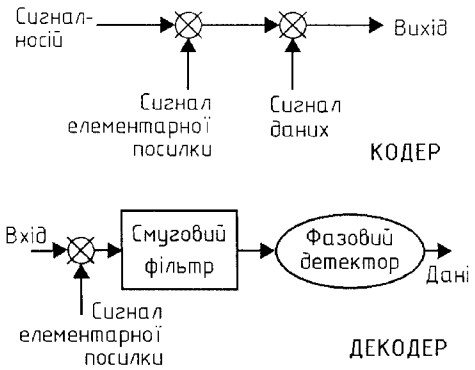


Рис. 5.87. Структурна схема кодека на основі розширення спектра

РСПП використовує двійкову фазову маніпуляцію, отож фаза чип-сигналу ПВП раз у раз чергується з фазою модульованої двійкової послідовності повідомлення (кодом) — рис. 5.88.

На стадії видобування фазові значення φ_0 та $\varphi_0+\pi$ інтерпретуються, відповідно, як біти «1» і «0», що ними кодувалася двійкова послідовність даних. При цьому передбачається наступне:

- псевдовипадковий ключ являє собою *M*-послідовність (він має максимально можливу кількість комбінацій, що рівномірно розподілені в

заданому діапазоні, і максимально довго не повторюються) і, таким чином, характеризується відносно плоским частотним спектром;

- на приймальному боці є відомим потік ключів для шифрування, дотримана синхронізація сигналу, а також відомі точки початку і кінця розширених даних;
- також на приймальному боці є відомими наступні параметри: частота слідування елементарних посилок, швидкість передавання даних и частота (вигляд) сигналу-носія.

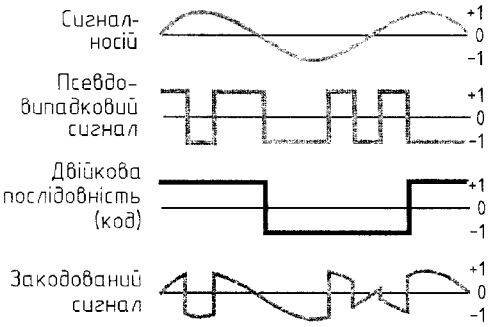


Рис. 5.88. Інформація, синтезована розширенням спектра і кодована методом прямої послідовності

На відміну від вищерозглянутого фазового кодування, метод РСПП вводить до звуку адитивний випадковий шум. Для того щоб тримати рівень шуму низьким і невідчутним на слух, розширений код послаблюється (без адаптації) приблизно до рівня 0,5% від динамічного діапазону звукового файлу-контейнера. Поєднання ж нескладної техніки повторення з корегувальним кодуванням гарантує цілісність коду повідомлення. Короткі сегменти двійкових кодових комбінацій об'єднуються і додаються до сигналу контейнера таким чином, щоб зменшити шуми перехідних процесів. Для цього під час декодування виконується усереднення по всьому сегменту. Зазначимо, що під час досліджень методу РСПП, авторами [14] була отримана швидкість передавання даних близько 4 бітів за секунду.

Наведемо приклад реалізації методу РСПП у середовищі *Mathcad*.

Крок 1

Початкові дані наступні: контейнер **C** := **READWAV**("C.wav"), властивості якого описані у п. 5.4.1. Повідомлення **M** := "© Alex Puzyrenko,

2017" довжиною $L_M := 8 \cdot \mathbf{strlen}(M) = 176$ бітів. Останнє вноситимемо до першого каналу контейнера: $\zeta := C^{\langle 1 \rangle}$.

Для вбудовування L_M-бітового повідомлення до контейнера, що має $i = \mathbf{rows}(\zeta) = 146994$ дискретних відліків, останній необхідно розбити на $N := \mathbf{trunc}(i / L_M) = 835$ сегментів, до кожного з яких вбудовуватиметься один біт повідомлення **M**. Таким чином, швидкість передавання даних становитиме:

$$f_д / N = 57.5 \text{ біт/с}.$$

Крок 2

Для кожного біта повідомлення необхідно згенерувати власну ПВП чип-сигналу з ±1, довжиною, як мінімум, **N** елементів. За основу генератора ПВЧ візьмемо ЛРЗЗЗ, описаний у підп. 5.3.3.3. У нашому випадку достатньою буде наступна кількість розрядів регістра:

$$d := \mathbf{ceil}(\log(N+1, 2)) = 10.$$

Період генерованої ПВП при цьому складатиме $2^d - 1 = 1023 > N$.

ПВП для різних значень **d** дозволяє одержувати програмний модуль (М.114), де вихідна послідовність визначається найменшими значущими бітами стану регістра: $\mathbf{CHIP}_i \leftarrow R_{bin_1}$. Керуючись достатністю, процес генерації триває до отримання **N**-го біта ПВП. На завершальному етапі одержана ПВП {0,1}, перетворюється на ПВП виду {-1,1}.

```
CHIP(seed) := | μ ← μ_d                                          (M.114)
              | R_dec_1 ← seed , R_bin ← D2B(R_dec_1 , d)
              | for i ∈ 2 .. N + 1
              |     CHIP_{i-1} ← R_bin_1
              |     bit ← 0
              |     for j ∈ 1 .. d
              |         bit ← R_bin_j ⊕ bit  if  μ_j
              |     R ← R_bin , R_bin_1 ← bit
              |     for j ∈ 2 .. d
              |         R_bin_j ← R_{j-1}
              |     R_dec_i ← B2D(R_bin)
              | 2 · CHIP - 1
```

Вагові коефіцієнти примітивного полінома, асоційованого з *M*-послідовністю, обираються, наприклад, з наступної множини:

$$\mu_1 := (1\ 1)^T$$
$$\mu_2 := (1\ 1\ 1)^T$$
$$\mu_3 := (1\ 0\ 1)^T$$
$$\mu_4 := (1\ 0\ 0\ 1\ 1)^T$$
$$\mu_5 := (1\ 0\ 1\ 0\ 0\ 1)^T$$
$$\mu_6 := (1\ 1\ 0\ 0\ 0\ 0\ 1)^T$$
$$\mu_7 := (1\ 1\ 0\ 0\ 0\ 0\ 0\ 1)^T$$
$$\mu_8 := (1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1)^T$$
$$\mu_9 := (1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1)^T$$
$$\mu_{10} := (1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1)^T$$
$$\mu_{11} := (1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1)^T$$
$$\mu_{12} := (1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1)^T$$

Крок 3

Безпосередньо вбудовування бітів повідомлення до аудіоконтейнера виконується програмним модулем (М.115). На початку даного модуля рядок M перетворюється на вектор значень {–1,1}, μ-й елемент якого за допомогою відповідної йому ПВП **CHİP**(μ) накладається на μ-й сегмент контейнера ς. Множник α обирається виходячи з вимог стійкості стеганосистеми і непомітності модифікації контейнера (рекомендуються значення α порядку однієї сотої).

$$\zeta' := \left|\begin{array}{l} M_{vec} \leftarrow \mathbf{str2vec}(M) \\ \mathbf{for}\ \ j \in 1..\mathbf{strlen}(M) \\ \quad M_{vec\_bin} \leftarrow \mathbf{if}\left(j=1, \mathbf{D2B}\left(M_{vec_1}, 8\right), \mathbf{stack}\left(M_{vec\_bin}, \mathbf{D2B}\left(M_{vec_j}, 8\right)\right)\right) \\ M_{vec\_bin} \leftarrow 2 \cdot M_{vec\_bin} - 1 \\ \mu \leftarrow 1 \\ \mathbf{while}\ \ \mu \le L_M \\ \quad \left|\begin{array}{l} \varsigma \leftarrow \mathbf{submatrix}[\zeta, [N\cdot(\mu-1)+1], N\cdot\mu, 1, 1] \\ \varsigma' \leftarrow \varsigma + \alpha\left[\overrightarrow{(\varsigma \cdot \mathbf{CHİP}(\mu))} \cdot M_{vec\_bin_\mu}\right] \\ \zeta' \leftarrow \mathbf{if}(\mu = 1, \varsigma', \mathbf{stack}(\zeta', \varsigma')) \\ \mu \leftarrow \mu + 1 \end{array}\right. \\ \zeta' \leftarrow \mathbf{stack}\left(\zeta', \mathbf{submatrix}\left(\zeta, L_M \cdot N + 1, \dot{I}, 1, 1\right)\right) \\ \zeta' \end{array}\right. \qquad \text{(М.115)}$$

Модифіковані сегменти ς' об'єднуються до спільного вектора ζ'. Після вбудовування останнього (L_M-го) біта повідомлення вектор ζ' подовжується на довжину контейнера ζ тими його кінцевими елементами, що не зазнали модифікації.

Приклад результату вбудовування для різних значень α зображено на рис. 5.89.

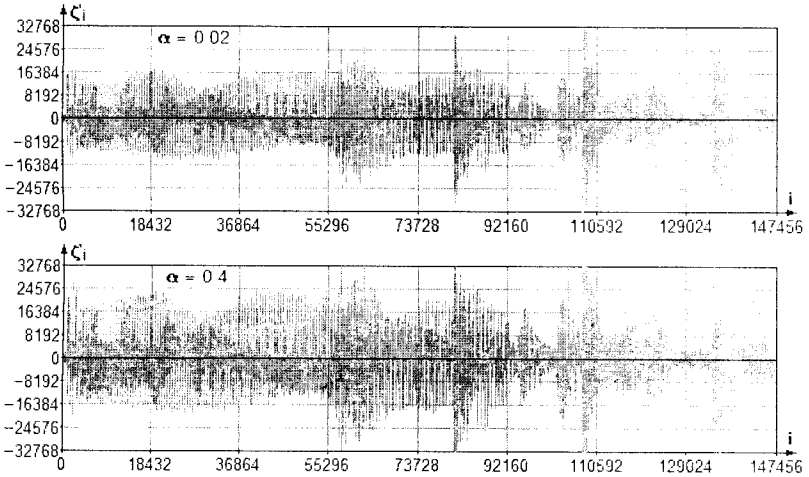


Рис. 5.89. Приклади часових діаграм першого каналу сигналу-контейнера у випадку внесення приховуваних даних шляхом РСПП при $\alpha = 0.02$ і $\alpha = 0.4$

Об'єднуємо масиви першого (модифікованого) і другого (немодифікованого) каналів до спільного масиву:

$$\mathbf{S} := \mathbf{augment}(\zeta', \mathbf{C}^{<2>})$$

і виконуємо запис отриманого при цьому результату до аудіофайлу:

$$\mathbf{WRITEWAV}(\text{"S\_DSSS.wav"}, \mathbf{f_Д}, \mathbf{Q}) := \mathbf{S}.$$

Крок 4

Процес видобування повідомлення полягає в наступному. Після імпортування аудіофайлу до робочого документа *Mathcad:*

$$\mathbf{S^*} := \mathbf{READWAV}(\text{"S\_DSSS.wav"}),$$

з масиву $\mathbf{S^*}$ виокремлюється той канал, до якого було проведено вбудовування (у нашому випадку — перший):

$$\zeta^* := S^{*<1>}.$$

Приймальна сторона повинна мати оригінальний аудіофайл, з якого також виокремлюється відповідний канал: $\zeta := C^{<1>}$. Відомими мають бути й кількість сегментів, на яку розбивається сигнал-носій: $N^* := 835$, і, звісно ж, ПВП: $CHiP = CHiP^*$.

Програмний модуль видобування вбудованого методом РСПП повідомлення — (М.116), — представлено нижче.

$$\begin{aligned}
M^* := \;& \mu \leftarrow 1 \\
& \textbf{while}\ \ \mu \cdot N^* \le i^* \\
& \quad \varsigma^* \leftarrow \textbf{submatrix}[\zeta^*, [N^* \cdot (\mu - 1) + 1], N^* \cdot \mu, 1, 1] \\
& \quad \varsigma \leftarrow \textbf{submatrix}[\zeta, [N^* \cdot (\mu - 1) + 1], N^* \cdot \mu, 1, 1] \\
& \quad c \leftarrow \overrightarrow{[(\varsigma^* - \varsigma) \cdot CHiP^*(\mu)]} \\
& \quad M^*_{bin_\mu} \leftarrow \textbf{if}(\textbf{sign}(\textbf{mean}(\varsigma)) = \textbf{sign}(\textbf{mean}(c)), 1, 0) \\
& \quad \mu \leftarrow \mu + 1 \\
& \textbf{for}\ \ j \in 1 .. \frac{\textbf{rows}(M^*_{bin})}{8} \\
& \quad M^*_{vec_j} \leftarrow B2D(\textbf{submatrix}(M^*_{bin}, 8 \cdot j - 7, 8 \cdot j, 1, 1)) \\
& \quad M^*_{vec_j} \leftarrow 32 \ \ \textbf{if}\ \ M^*_{vec_j} < 32 \\
& \textbf{vec2str}(M^*_{vec})
\end{aligned} \tag{М.116}$$

Якщо до сегмента було вбудовано «1», то середні значення оригінального (**c**) і модифікованого (**c'**) сегментів матимуть однакові знаки (функція **sign**(...)), якщо ж «0» — різні.

Недоліком такої системи можна назвати необхідність наявності в одержувача оригінального (немодифікованого) аудіофайлу. Можливим виходом з цієї ситуації є, наприклад, запис модифікованого сегмента першого каналу до відповідної ділянки паралельного (другого) каналу, тоді як оригінальний перший канал залишатиметься без змін:

$$S := \textbf{augment}(C^{<1>}, \textbf{putregion}(C^{<2>}, \zeta', a, b)).$$

Таким чином, одночасно передаватимуться як пустий, так і заповнений контейнери. На приймальному боці зазначена особливість вбудовування може бути врахована наступною зміною відповідних рядків у (М.116):

$$\begin{aligned}
& \varsigma^* \leftarrow \textbf{submatrix}[\zeta^{*(2)}, [N^* \cdot (\mu - 1) + 1], N^* \cdot \mu, 1, 1] \\
& \varsigma \leftarrow \textbf{submatrix}[\zeta^{*(1)}, [N^* \cdot (\mu - 1) + 1], N^* \cdot \mu, 1, 1]
\end{aligned}$$

Певна річ, модифікувати вказаним чином весь стереофонічний аудіосигнал є недоцільним — через повне зникнення стереоефекту. Тому, приміром, можна обмежитися однією чи декількома заздалегідь обумовленими з приймальною стороною ділянками або навіть окремими відліками по всій тривалості сигналу (наприклад, домовитися про використання алгоритму псевдовипадкового їх обрання). На основі зазначених ділянок (відліків) створюється вибірка, в яку або з якої й виконуватиметься вбудовування/видобування бітів повідомлення.

Крок 5

Обчислені показники звукового спотворення аудіоконтейнера вбудовуванням повідомлення з розширеним спектром занесено до табл. 5.19 (стор. 302).

5.4.4. Метод кодування луно-сигналу

Даний метод розуміє під собою вбудовування даних до аудіосигналу-контейнера шляхом введення до нього луно-сигналу [14]. Дані приховуються зміною в останнього трьох параметрів: початкової амплітуди, швидкості загасання [*(початкова амплітуда – загасання)/δ*] і зсуву (див. рис. 5.90).

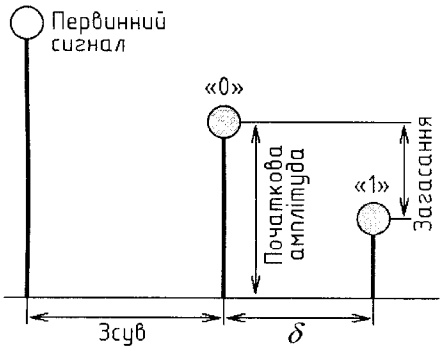


Рис. 5.90. Регульовані параметри луно-сигналу

Коли зсув (або затримка) між первинним та луно-сигналом зменшується, два сигнали змішуються і, починаючи з деякого значення затримки, ССЛ стає вже не спроможною виявити різницю між ними, а луно-сигнал сприймається просто як додатковий резонанс. Значення

згаданого зсуву важко визначити точно, оскільки воно залежить від якості первинного аудіозапису, типу звуку, для якого формується луно-сигнал, і, зрештою, — від слухача.

У загальному випадку, автори [14] дійшли висновку, що для більшості звуків і більшості слухачів «злиття» відбувається при затримці близько однієї мілісекунди.

Двійковий стеганокодер використовує два часи затримки: один для відображення «0» (*зсув*), а інший — для відображення «1» (*зсув*+$\boldsymbol{\delta}$). Обидва значення затримки є меншими за той граничний час, при якому ССЛ усе ще здатна розпізнати луно-сигнал. Крім зменшення часу затримки, для забезпечення непомітності також можна встановити рівні початкової амплітуди і часу загасання таким чином, щоб вони також не перевищували поріг чутливості ССЛ.

Процес вбудовування даних може бути представлений у вигляді пристрою, що реалізує одну з двох можливих системних функцій. У часовій області системні функції — це дискретні у часі показникові функції (рис. 5.91), що відрізняються одна від одної лише затримкою між імпульсами (відліками), що їх складають.

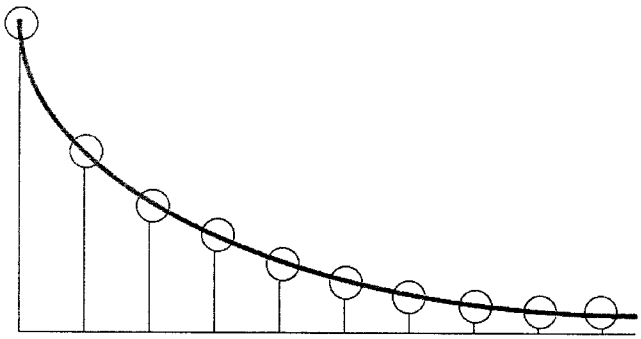


Рис. 5.91. Дискретна у часі експоненційна функція

Розглянемо приклад з двома імпульсами (один для копіювання первинного сигналу, а інший — для формування луно-сигналу). Очевидно, що збільшення кількості імпульсів призведе до зростання кількості луно-сигналів.

На рис. 5.92 представлено системні функції для кодування двійкових «1» і «0». Обробка сигналу у відповідності до рис. 5.92, *а* або *б* матиме своїм результатом закодований сигнал — рис. 5.93.

Затримка $\boldsymbol{\delta}_{бій}$ між первинним і луно-сигналом є залежною від того, яке відображення (системна функція) було використане. Відображення

«1» створюється затримкою у δ_1 секунд, тоді як відображення «0» — затримкою у δ_0 секунд.

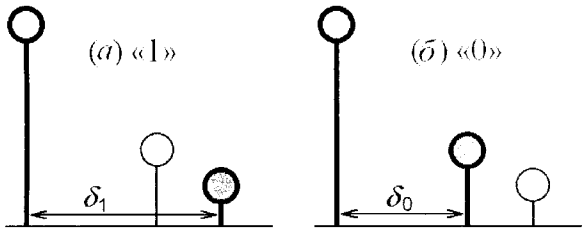


Рис. 5.92. Двійковий спосіб відображення луно-сигналу

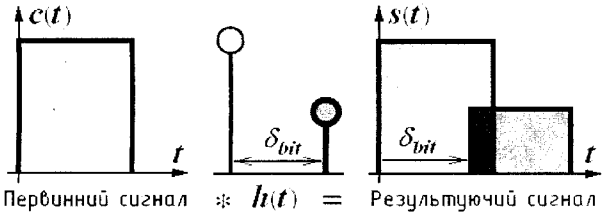


Рис. 5.93. Приклад луно-відображення

Для того щоб до первинного сигналу вбудувати більше одного біта, він розкладається на менші сегменти, кожен з яких розглядається як окремий сигнал, що до нього шляхом луно-відображення може бути вбудовано один біт приховуваного повідомлення. Підсумковий аудіосигнал (контейнер-результат, утримуючий певну кількість бітів) являє собою нове об'єднання всіх незалежно кодованих сегментів первинного сигналу.

На рис. 5.94 наведено приклад, при якому сигнал було розділено на сім однакових сегментів, позначених як ***a***, ***b***, ***c***, ***d***, ***e***, ***f*** та ***g***.

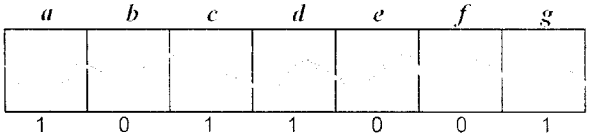


Рис. 5.94. Приклад розбиття первинного аудіосигналу на менші сегменти для вбудовування повідомлення, що являє собою двійкову послідовність

Нехай необхідно, щоб сегменти ***a***, ***c***, ***d*** та ***g*** містили у собі «1». Отже, для кожного з них треба застосувати системну функцію відображення одиниці (див. рис. 5.92, *а*). Кожен сегмент індивідуально згортається з системною функцією. Нулі, вбудовані до сегментів ***b***, ***e*** та ***f***, кодуються аналогічним чином, використовуючи спосіб відображення нуля (рис. 5.92, *б*). Одержані після згортання з відповідною системною функцією сегменти-результати наново об'єднуються до спільного сигналу.

Для досягнення мінімальної помітності повторного з'єднання, у [14] пропонується попередньо створити окремі «одиничний» і «нульовий» луно-сигнали, повторюючи первинний і використовуючи відповідні відображення «1» і «0». Приклад отриманих у підсумку сигналів зображено на рис. 5.95.

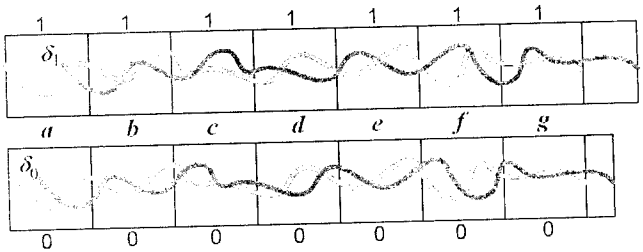


Рис. 5.95. Приклад створення «одиничного» і «нульового» луно-сигналів

«Одиничний» і «нульовий» луно-сигнали містять, відповідно, самі лише одиниці і нулі. Для того щоб об'єднати ці два сигнали, також створюються два так званих змішувальних сигнали (рис. 5.96), що являють собою послідовність двійкових даних, стан якої залежить від того, який біт треба приховати у тому чи іншому сегменті первинного сигналу.

«Одиничний» і «нульовий» змішувальні сигнали помножуються на відповідні їм луно-сигнали. Іншими словами, останні масштабуються одиницею або нулем протягом усього часу існування сигналу в залежності від того, який біт передбачається вбудувати до будь-якого з його окремих сегментів. У подальшому два результати додаються один до одного.

Не зайвим буде зауважити, що «нульовий» змішувальний сигнал являє собою інверсію «одиничного» змішувального сигналу. Крім того, фронти переходів кожного сигналу є положистими, а сума обох змішувальних сигналів завжди дорівнює одиниці. У комплексі все це дозволяє одержати плавний перехід між сегментами, кодованими різ-

ними бітами, а також запобігає виникненню різких перепадів у звучанні результуючого (змішаного) аудіосигналу.

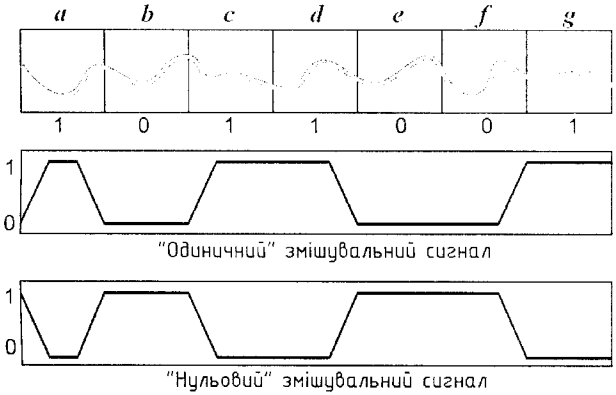


Рис. 5.96. Змішувальні сигнали

Блок-схема, що представляє повний процес вбудовування шляхом кодування луно-сигналів, зображена на рис. 5.97.

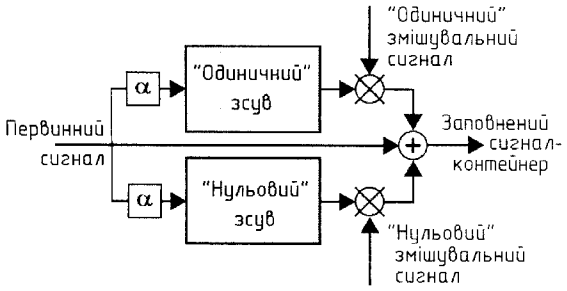


Рис. 5.97. Блок-схема вбудовування кодуванням луно-сигналів

Видобування вбудованої інформації розуміє під собою виявлення інтервалу між луно-сигналами окремих сегментів. Для цього необхідно дослідити у двох зумовлених відповідними зсувами позиціях амплітуду автокореляційної функції (АКФ) косинус-перетворення Фур'є натурального логарифма спектра потужності (або так званого кепстра) сигналу-контейнера [92, 93, 124, 125]:

$$ac[u] = \dot{I}FT\left\{ln\left(\left|FT(s[n])\right|^2\right)\right\}, \qquad (5.58)$$

де $s[n]$ — вектор амплітуд часових відліків аналізованого сегмента сигналу-контейнера **S**; $FT(...)$ та $\dot{I}FT(...)$ — функції, відповідно, прямого і зворотного перетворень Фур'є; $ac[u]$ — результуючий вектор кепстра сигналу $s[n]$.

Розглянемо особливості процесу видобування, наведені у [14]. Нехай одержано закодований сигнал, що являє собою таку послідовність імпульсів, в якій останні відокремлені один від одного визначеним інтервалом і характеризуються експоненціальним загасанням амплітуди; в усіх інших точках сигнал дорівнює нулеві (рис. 5.98).

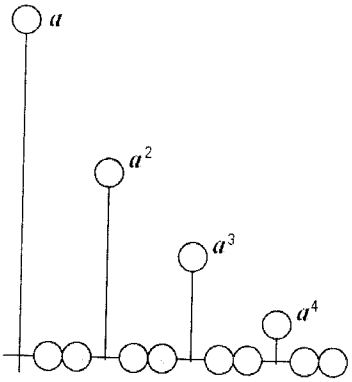


Рис. 5.98. Приклад сигналу $s[n] = a^n \cdot x[n]$: $(0 < a < 1)$

Наступним кроком є пошук кепстра луно-версії сигналу. Результат обчислення кепстра робить інтервал між луно-сигналом і первинним сигналом дещо більш виразним.

Нажаль, результат обчислення кепстра до того ж ще й дублює луно-сигнал через кожні δ секунд. На рис. 5.99 це зображено послідовністю імпульсів на виході кепстрального обчислювача. Крім того, амплітуди імпульсів, що представляють луно-сигнали, є малими по відношенню до первинного сигналу. Як наслідок, їх важко виявити.

Розв'язок даної проблеми полягає в обчисленні АКФ кепстра. За допомогою однократного відображення сигналу із затримкою δ (рис. 5.100), отримуємо результат, зображений на рис. 5.101. При цьому значно посиленим є лише перший імпульс, оскільки його «підтримано» наступними за ним імпульсами. Таким чином, у позиції першого імпульсу ми отримуємо сплеск. Подібно до першого імпульсу, сплеск

повторюється через δ_1 або δ_0 секунд після сплеску первинного сигналу. Залишкові ж складові імпульсів прямують до нуля, що, у підсумку, дозволяє ефективно послаблювати шуми.

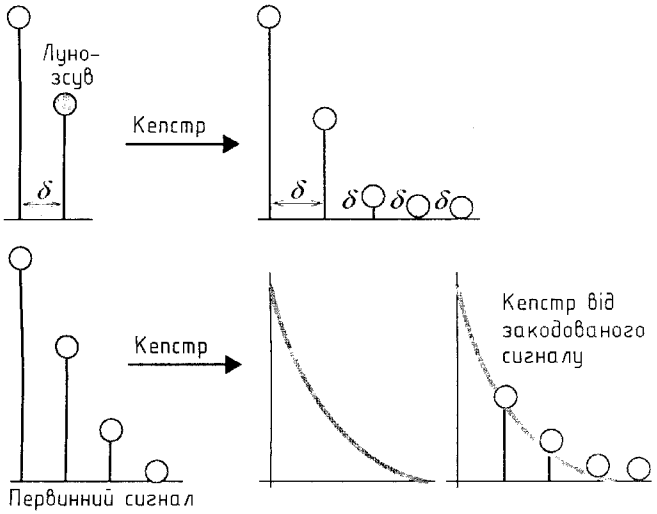


Рис. 5.99. Процес отримання кепстра від луно-кодованого сигналу

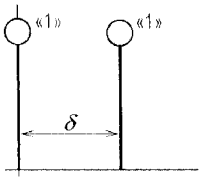


Рис. 5.100. Принцип відображення сигналу

Розглянемо критерій прийняття рішення стосовно того, який біт («1» чи «0») було приховано у часовій затримці між первинним сигналом і затримкою δ перед сплеском АКФ. Згадаємо, що «1» кодувалася розміщенням луно-сигналу через δ_1, а «0» — через δ_0 секунд після оригіналу. При видобуванні, якщо значення АКФ через δ_1 секунд є більшим, аніж через δ_0 секунд, то поточному біту формованої копії повідомлення M^* ставиться у відповідність одиниця. В іншому випадку — бітові присвоюється нуль.

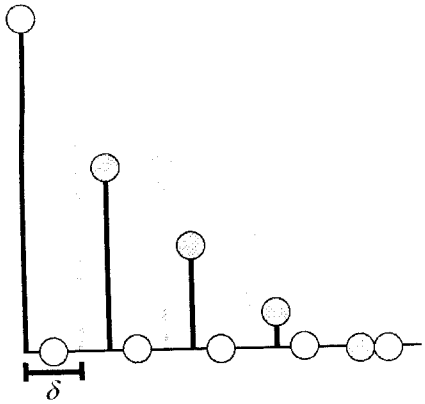


Рис. 5.101. Результат однократного луно-відображення сигналу

За твердженням авторів [14], за допомогою даного методу цілком можливо приховувати і видобувати інформацію у вигляді двійкового коду до/з потоку аудіоданих з мінімальною зміною первинного сигналу при ППЗ, приблизно, у 16 біт/с. Під мінімальною зміною розуміється той факт, що середньостатистична людина не відчуватиме суттєвої різниці між первинним і модифікованим сигналами. Проте, у своїй же роботі [108] автори вказують на те, що запропонований ними метод не є універсальним — для деяких типів аудіосигналів неможливо одержати достатньо високий коефіцієнт вірно розпізнаних при видобуванні бітів — навіть за відсутності у каналі зв'язку завад.

Розглянемо реалізацію методу луно-кодування за допомогою програми *Mathcad*.

Крок 1

Початкові дані:

- аудіоконтейнер з частотою дискретизації $f_д$ = 48000 Гц і кількістю рівнів квантування **Q** = 16: C_{total} := **READWAV**("C.wav"); **C** := $C_{total}^{<1>}$, **rows**(**C**) = 146994 відліки;
- повідомлення **M** := "© Alex Puzyrenko, 2017" довжиною N_M := **strlen**(**M**) = = 22 символи або $L_M := 8 \cdot N_M$ = 176 бітів.

Крок 2

Нехай пульова затримка між первинним і луно-сигналом складає δ_0 := 25 дискретних відліків (або $\delta_0 / f_д$ = 0.521 мс), а одинична — δ_1 := 52 відліки (або $\delta_1 / f_д$ = 1.083 мс).

«Одиничний» і «нульовий» луно-сигнали одержимо шляхом звичайного зміщення на $\delta_{0(1)}$ відліків елементів контейнера-оригіналу з наступним поелементним підсумовуванням одержаних векторів (попередньо помножених на коефіцієнт загасання α) з вектором **C** — програмні модулі (М.117) і (М.118).

$$\mathbf{C_0} := \left| \begin{array}{l} \textbf{for } i \in 1..\delta_0 \\ \quad C0_i \leftarrow 0 \\ C_0 \leftarrow \alpha \cdot \mathbf{stack}(C_0, C) \end{array} \right. \qquad \mathbf{C_1} := \left| \begin{array}{l} \textbf{for } i \in 1..\delta_1 \\ \quad C1_i \leftarrow 0 \\ C_1 \leftarrow \alpha \cdot \mathbf{stack}(C_1, C) \end{array} \right. \qquad \text{(М.117)}$$

$$\Sigma C_0 := \left| \begin{array}{l} \textbf{for } i \in 1..\mathbf{rows}(C) \\ \quad \Sigma C0_i \leftarrow C_i + C0_i \\ \Sigma C_0 \end{array} \right. \qquad \Sigma C_1 := \left| \begin{array}{l} \textbf{for } i \in 1..\mathbf{rows}(C) \\ \quad \Sigma C1_i \leftarrow C_i + C1_i \\ \Sigma C_1 \end{array} \right. \qquad \text{(М.118)}$$

Перевіряємо визначальні розмірності:

$\mathbf{rows}(C_0) = 147019$; $\mathbf{rows}(C_1) = 147046$.

$\mathbf{rows}(C_0) - \mathbf{rows}(C) = 25$; $\mathbf{rows}(C_1) - \mathbf{rows}(C) = 52$.

$\mathbf{rows}(\Sigma C_0) = 146994$; $\mathbf{rows}(\Sigma C_1) = 146994$.

Фрагменти (перші 140 відліків) результату зсуву сигналу **C** на δ_0 і δ_1 часових відліків зображено на рис. 5.102.

Крок 3

Для можливості приховання більше одного біта даних первинний сигнал **C** необхідно розділити на менші за розміром сегменти, кожен з яких розглядатиметься як окремий сигнал і до якого шляхом луно-відображення може бути вбудовано необхідний біт.

Обчислимо кількість відліків в одному такому сегменті, виходячи з бітової довжини L_M повідомлення, округлюючи результат до найближчого найменшого цілого:

$$N_Б := \mathbf{trunc}(\mathbf{rows}(C)/L_M) = 835 \text{ відл./сегм.}$$

Крок 4

Виходячи зі вказаної при описі методу необхідності у похилених фронтах імпульсів змішувальних сигналів (трапецеїдальні імпульси), попередньо задаємося наступним:

- розмах імпульсу: $U := 1$ В;
- тривалість кожного з фронтів: $\tau := 120$ відліків;
- тривалість імпульсу на рівні **U**: $T := N_Б - \tau = 715$ відліків.

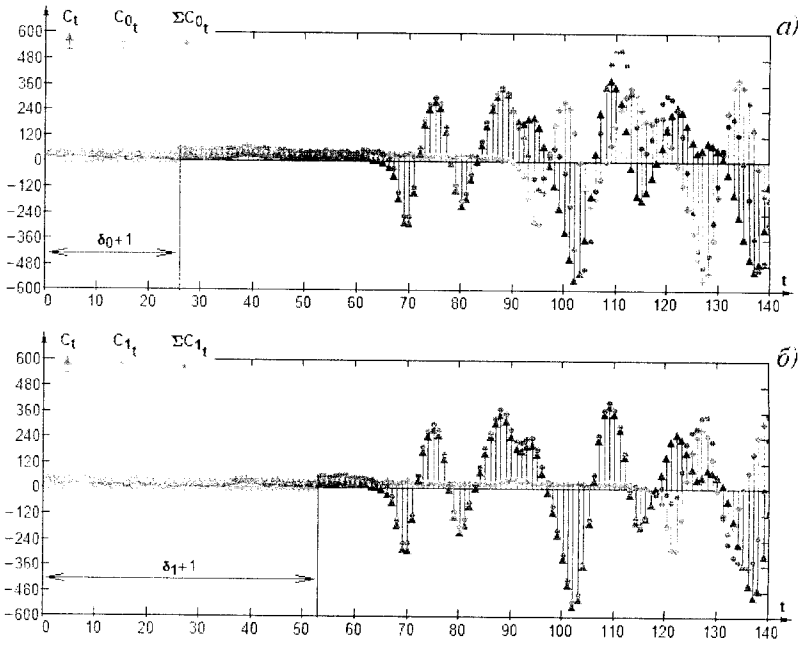


Рис. 5.102. «Нульовий» (*а*) та «одиничний» (*б*) луно-сигнали сигналу **C** при $\alpha = 1$

Амплітуди відліків імпульсу формуємо за допомогою складових програмного модуля (М.119).

$$\text{пф} := \left| \begin{array}{l} \text{for } n \in 1..\tau \\ \quad \text{пф}_n \leftarrow \text{if}\left(\tau > 0,\ \frac{n-1}{\tau},\ 0\right) \\ \text{пф} \end{array} \right. \qquad \text{(М.119)}$$

$$u := \left| \begin{array}{l} \text{for } n \in (\tau+1)..(\tau+T) \\ \quad u_{n-\tau} \leftarrow U \\ u \end{array} \right.$$

$$\text{зф} := \left| \begin{array}{l} \text{for } n \in (\tau+T+1)..(2\cdot\tau+T) \\ \quad \text{зф}_{n-(\tau+T)} \leftarrow \text{if}\left[\tau > 0,\ \frac{n-(2\cdot\tau+T)}{-\tau},\ 0\right] \\ \text{зф} \end{array} \right.$$

$$I := \text{stack}(\text{пф}, u, \text{зф})$$

Графічну інтерпретацію обчислення модуля (М.119) зображено на рис. 5.103.

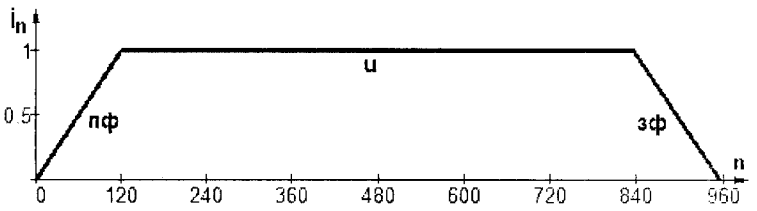


Рис. 5.103. Трапецеїдальний імпульс **i**, побудований за $2\cdot\tau+T$ відліками

Формування змішувальних сигналів виконаємо, користуючись програмним модулем (М.120). При цьому приймаємо, що кожен біт повідомлення кодується половиною переднього фронту (**пф**) імпульсу, його одиничним рівнем (**u**) та половиною заднього фронту (**зф**):

$$\tau/2+T+\tau/2=835 \text{ відл.}$$

(М.120)

```
μ1 := | M_vec ← str2vec(M)
      | for j ∈ 1.. N_M
      |   M_vec_bin ← if[j = 1, D2B[M_vec_1, 8], stack[M_vec_bin, D2B[M_vec_j, 8]]]
      | μ ← stack[0·submatrix[пф, round(τ/2)+1, τ, 1, 1]+1, u]  if M_vec_bin_1
      | μ ← stack[0·submatrix[зф, round(τ/2)+1, τ, 1, 1], 0·u]  otherwise
      | for m ∈ 2.. L_M
      |   if M_vec_bin_m
      |     μ ← stack(μ, пф, u)  if M_vec_bin_m ≠ M_vec_bin_m-1
      |     μ ← stack[μ, (0·пф+1), u]  otherwise
      |   otherwise
      |     μ ← stack(μ, зф, 0·u)  if M_vec_bin_m ≠ M_vec_bin_m-1
      |     μ ← stack(μ, 0·зф, 0·u)  otherwise
      | while rows(μ) ≤ rows(C)
      |     μ ← if[M_vec_bin_L_M, stack[μ, (0·пф+1), u], stack(μ, 0·пф, 0·u)]
      | μ
      →
μ0 := |−1·μ1+1|
```

Після формування змішувального сигналу для всіх L_M бітів повідомлення, він дописується до остаточної довжини **rows(C)** нульовим

або одиничним рівнем — у залежності від значення останнього (L_M-го) біта. При цьому нульовий змішувальний сигнал (μ_0) одержується з одиничного (μ_1) з урахуванням того, що їх сума має дорівнювати одиниці.

Змішувальні сигнали для першої вісімки бітів обраного нами повідомлення **M** (*ASCII*-код символу «©», записаний у зворотному порядку — від молодших бітів до старших) представлено на рис. 5.104.

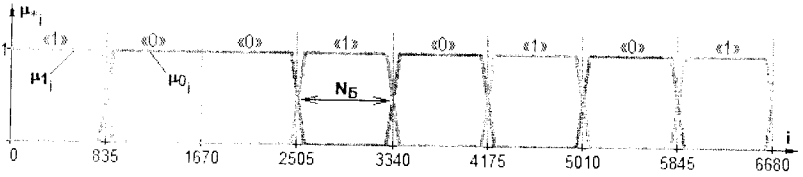


Рис. 5.104. Приклад «нульового» (μ_0) і «одиничного» (μ_1) змішувальних сигналів

Крок 5

Безпосередньо вбудовування бітів повідомлення до аудіоконтейнера виконує програмний модуль (М.121).

```
S := | M_vec ← str2vec(M)                                                    (M.121)
     | for j ∈ 1..N_M
     |   M_vec_bin ← if[j = 1, D2B[M_vec_1, 8], stack[M_vec_bin, D2B[M_vec_j, 8]]]
     | for m ∈ 1..L_M
     |   if M_vec_bin_m
     |     | Σ ← submatrix[ΣC_1, [N_Б·(m − 1) + 1], N_Б·m, 1, 1]
     |     | μ ← submatrix[μ_1, [N_Б·(m − 1) + 1], N_Б·m, 1, 1]
     |   otherwise
     |     | Σ ← submatrix[ΣC_0, [N_Б·(m − 1) + 1], N_Б·m, 1, 1]
     |     | μ ← submatrix[μ_0, [N_Б·(m − 1) + 1], N_Б·m, 1, 1]
     |   for n ∈ 1..N_Б
     |     S'_n ← Σ_n·μ_n
     |   S ← if(m = 1, S', stack(S, S'))
     | for i ∈ rows(S)+1..rows(C)  if rows(S) < rows(C)
     |   S_i ← C_i
     | scale[S, −2^(Q−1), 2^(Q−1)]
```

У залежності від значення поточного біта повідомлення — $\mathbf{M}_{vec\_bin_m}$ — проводиться виокремлення сегмента заданої розмірності ($\mathbf{N_Б}$) з відповідного луно-сигналу ($\Sigma\mathbf{C_1}$ або $\Sigma\mathbf{C_0}$) та зі змішувального сигналу (μ_1 або μ_0), що у подальшому поелементно перемножуються (для цього можна також використати й операцію векторизації).

Отримані для кожного біта вектори **S'** формують собою загальний вектор заповненого контейнера **S**, у кінець якого після вбудовування останнього біта повідомлення дописуються елементи контейнера-оригіналу **C**, що не зазнали модифікації. Очевидно, що кількість елементів сформованого вектора **S** відповідатиме аналогічній характеристиці для **C**: **rows**(**S**) = 146994.

Приклад формування перших $8 \cdot \mathbf{N_Б}$ відліків заповненого контейнера наведено на рис. 5.105. Часова діаграма заповненого контейнера наведена на рис. 5.106.

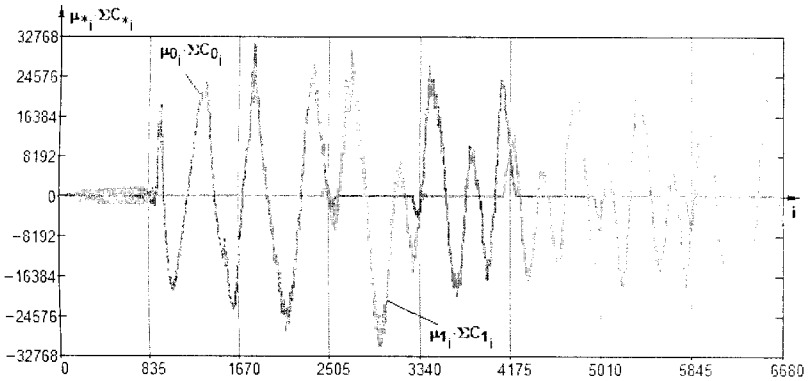


Рис. 5.105. Приклад формування контейнера-результату

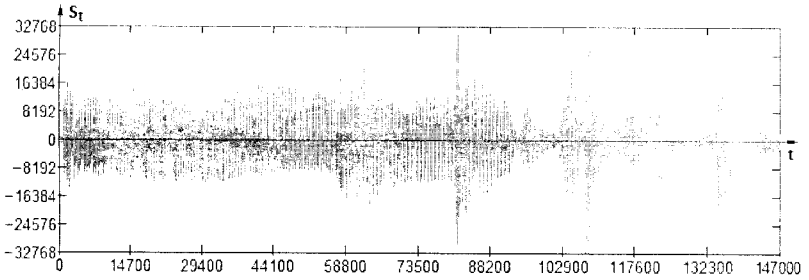


Рис. 5.106. Часова діаграма заповненого методом луно-кодування аудіоконтейнера

Одержаний у підсумку вектор **S** об'єднуємо з немодифікованим другим каналом і записуємо результат до файлу:

$$\mathbf{WRITEWAV}(\text{"S\_echo.wav"}, f_д, Q) := \mathbf{augment}(\mathbf{S}, \mathbf{C_{total}}^{\langle 1 \rangle}).$$

Крок 6

Для видобування прихованого повідомлення передбачається наступне: одержувачеві відомі розмірність блоків, на які розбивається контейнер ($N^*_Б := N_Б$), значення нульової та одиничної затримок ($\delta^*_0 := \delta_0$, $\delta^*_1 := \delta_1$).

Програмний модуль видобування даних — (М.122). До його основи покладено обчислення автокореляційної функції кепстра (5.58), а для більш надійного видобування аналізується окіл відліків δ^*_0 і δ^*_1.

$\mathbf{M^*} :=$ | $m \leftarrow 1$ (М.122)

while $N^*_Б \cdot m < \mathbf{rows}(S^*)$

$s \leftarrow \mathbf{submatrix}\left[S^*, \left[N^*_Б \cdot (m-1)+1\right], N^*_Б \cdot m, 1, 1\right]$

$s \leftarrow \mathbf{stack}(s, 0 \cdot s)$

$ac \leftarrow \mathbf{mag}(\mathbf{icfft}(2 \cdot \ln(\mathbf{mag}(\mathbf{cfft}(s)))))$

$\text{«0»} \leftarrow \sum_{j=\delta^*_0-1}^{\delta^*_0+1} ac_j \quad , \quad \text{«1»} \leftarrow \sum_{j=\delta^*_1-1}^{\delta^*_1+1} ac_j$

$M^*bin_m \leftarrow \mathbf{if}(\text{«1»} > \text{«0»}, 1, 0)$

$m \leftarrow m + 1$

for $j \in 1 .. \frac{\mathbf{rows}(M^*_{bin})}{8}$

$M^*vec_j \leftarrow \mathbf{B2D}(\mathbf{submatrix}(M^*_{bin}, 8 \cdot j - 7, 8 \cdot j, 1, 1))$

$M^*vec_j \leftarrow 32 \quad \mathbf{if} \quad M^*vec_j < 32$

$\mathbf{vec2str}(M^*vec)$

Приклади результатів обчислення АКФ кепстра для 1-го і 2-го бітів повідомлення наведено на рис. 5.107.

Результат видобування:

$$\mathbf{M^*} = \text{"© Alex Puzyrenko, 2017"}$$

Крок 7

Результати обчислення показників звукового спотворення контейнера при вбудовуванні до нього даних шляхом луно-кодування зведено до табл. 5.19.

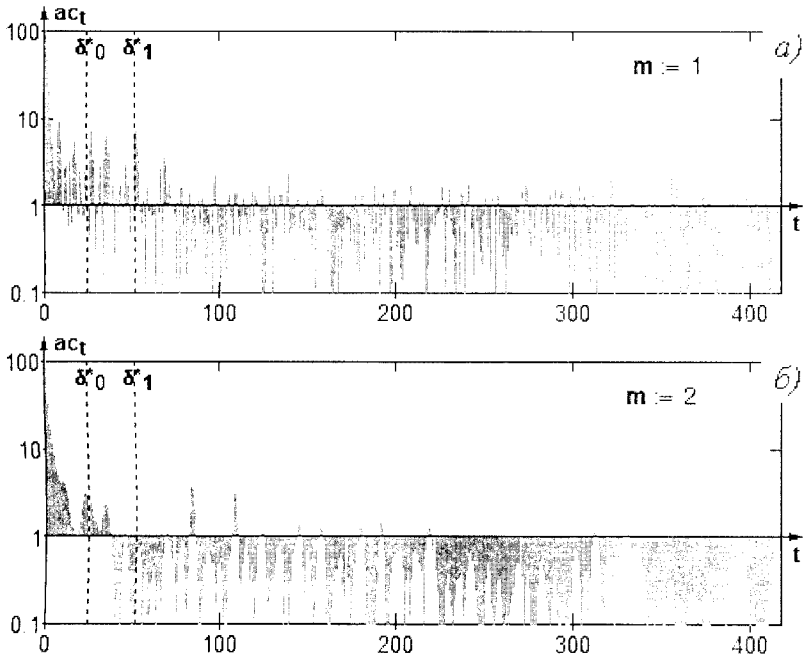


Рис. 5.107. Приклад вигляду АКФ кепстра сигналу, що містить одиничне (*а*) і нульове (*б*) луно-відображення

5.4.5. Метод стеганокодування стиснутих із втратами аудіосигналів

В якості прикладу організації КППД на основі аудіоконтейнерів, що являють собою результат стиснення із втратами звукових сигналів, розглянемо спосіб перенесення додаткової інформації (ДІ) стеганоканалами, створеними на основі радіопрограмних потоків субсмугових відліків *MPEG Audio Layer II (Musicam)* у складі цифрового звукового мовлення (ЦЗМ) *«Eureka-147»/DAB* (*Digital Audio Broadcasting*) [138,139]. При цьому для частот дискретизації 48 і 24 кГц використовуються, відповідно, стандарти *ISO/IEC 11172-3* [126,127] та *ISO/IEC 13818-3* [128,129]. *Musicam* дозволяє зберігати аудіодані з імпульсно-кодовою модуляцією (ІКМ) у форматі 44.1 кГц/16 біт/стерео без відчутних втрат якості при швидкості цифрового потоку 192-256 кбіт/с, що становить 6...8-кратний виграш і відповідає рекомендаціям [130]. Концептуальні засади аудіостиснення *Musicam* представлені у Додатках А і Б.

Таблиця 5.19

Показники спотворення у випадку приховування даних в аудіосередовищі

Назва показника спотворення	*Оригінал*	*Методи приховування*			
		Заміна НЗБ з ПВ інтервалом	*Фазове кодування*	*Розширення спектра ($\alpha = 0.02$)*	*Кодування луно-сигналу*
Кількість бітів у повідомленні, L_M	—	176+56	176	176	176
Кількість модифікованих відліків контейнера	0	103	146994	146942	146994
Максимальна абсолютна різниця, *MD*	0	1	5575	655	30300
Середня абсолютна різниця, *AD*	0	$3.504 \cdot 10^{-4}$	41.093	47.321	$1.912 \cdot 10^{3}$
Нормована середня абсолютна різниця, *NAD*	0	$7.612 \cdot 10^{-8}$	$8.928 \cdot 10^{-3}$	$1.028 \cdot 10^{-2}$	0.404
Середньоквадратична помилка, *MSE*	0	$3.504 \cdot 10^{-4}$	$1.656 \cdot 10^{4}$	$8.024 \cdot 10^{3}$	$7.416 \cdot 10^{6}$
Нормована середньоквадратична помилка, *NMSE*	0	$9.130 \cdot 10^{-12}$	$4.315 \cdot 10^{-4}$	$2.019 \cdot 10^{-1}$	0.185
L^p-норма, при $p = 2$	0	0.019	128.670	89.576	$2.723 \cdot 10^{3}$
Відношення сигнал/шум, *SNR*	∞	$1.095 \cdot 10^{11}$	$2.318 \cdot 10^{3}$	$4.782 \cdot 10^{3}$	5.41
Максимальне відношення сигнал/шум, *PSNR*	∞	$3.065 \cdot 10^{12}$	$6.486 \cdot 10^{4}$	$1.338 \cdot 10^{5}$	144.782
Якість звучання, *AF*	1	1.000000	0.999569	0.999791	0.815146
Нормована взаємна кореляція, *NC*	1	1.000000	0.999784	0.999970	0.836138
Якість кореляції, *CQ*	8337.127	8337.127	8335.328	8336.880	7088.592
Структурний зміст, *SC*	1	1.000000	1.000000	0.999850	1.166683
Загальне сигма-відношення сигнал/шум, *GSSNR*	∞	$3.501 \cdot 10^{16}$	$4.727 \cdot 10^{14}$	$1.777 \cdot 10^{8}$	181.669
Сигма-відношення сигнал/шум, *SSNR*	∞	165.441	146.746	82.497	22.593
Нормоване відношення сигнал/помилка, *NSER*	∞	$1.095 \cdot 10^{11}$	$2.318 \cdot 10^{3}$	$4.782 \cdot 10^{3}$	5.41
Подібність гістограм, *HS*	0	206	71983	73153	73701

Блок-схеми стеганографічного аудіокодека (рис. 5.108 і 5.109) є результатом розширення на операції стеганографічного кодування і декодування субсмугових відліків блок-схем аудіокодування і декодування *MPEG* (див. Додаток А, рис. А.4 і А.6).

Рис. 5.108. Блок-схема операцій стеганокодування аудіопотоку *MPEG*

Рис. 5.109. Блок-схема операцій стеганокодування аудіопотоку *MPEG*

Крок 1

Імпортуємо амплітуди часових дискретних відліків вхідного ІКМ-аудіосигналу, представленого наразі у типовому, нестиснутому *WAV*-форматі:

PCM = **READWAV**("C.wav"),

Нехай при цьому кількість каналів $\mathbf{N_к} = 1$, частота дискретизації $\mathbf{f_д} = 48$ кГц, розрядність кодування відліків $\mathbf{\Delta A} = 16$ біт/відл.; середня швидкість обробки відтворювачем у реальному часі **VB** = 96 кБ/с.

Інші вихідні дані є наступними:

- загальна кількість відліків: $\mathbf{L} := \mathbf{rows}(\mathbf{PCM}) = 146994$ відл.;
- інтервал дискретизації: $\Delta\mathbf{t} := 1/\mathbf{f_д} = 20.833$ мкс;
- загальна тривалість аудіосигналу: $\mathbf{t_\Sigma} := \mathbf{L}\cdot\Delta\mathbf{t} = 3.062$ с;
- розмір файлу "C.wav": $\varphi := \mathbf{L}\cdot\Delta\mathbf{A} = 2.352$ Мб (або 294 кБ);
- кількість відліків в окремому аудіокадрі *MPEG:* $\mathbf{S_K} := 1152$ відл.;
- тривалість окремого аудіокадру *MPEG:* $\mathbf{t_K} := \mathbf{S_K}\cdot\Delta\mathbf{t} = 24$ мс;
- загальна кількість аудіокадрів: $\mathbf{N_{A\Sigma}} := \mathbf{trunc}(\mathbf{L}/\mathbf{S_K}) = 127$ кадрів;
- кількість аудіокадрів, обрана для розгляду: $\mathbf{N_A} := 127$ кадрів;
- тривалість обраного сегмента аудіосигналу: $\mathbf{t_C} := \mathbf{N_A}\cdot\mathbf{t_K} = 3.048$ с.

Оберемо сегмент з $\mathbf{N_A}$ кадрів (рис. 5.110)[XXI]:

$$\mathbf{PCM} := \mathbf{submatrix}(\mathbf{PCM}, 0, \mathbf{S_K}\cdot\mathbf{N_A} - 1, 0, 0).$$

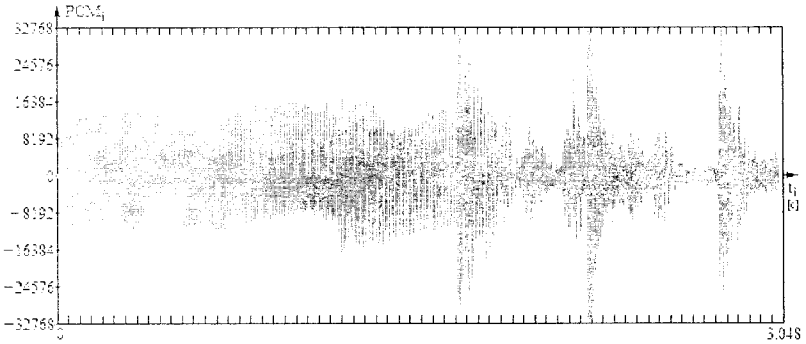


Рис. 5.110. Часова діаграма ІКМ-відліків окремого каналу аудіосигналу контейнера

Крок 2

Субсмуговий аналіз у БФА (рис. А.3 Додатка) здійснюється розщепленням вхідного широкосмугового сигналу, що має частоту дискретизації $\mathbf{f_д} = 48$ кГц, на 32 послідовні рівновіддалені субсмугові складові (фільтровані відліки) з частотою дискретизації $\mathbf{f_д}/32 = 1500$ Гц для кожної. Згідно [126, 127], крім інших проміжних структур даних БФА використовує ще й буфер $\mathbf{X}$ — чергу типу *FIFO* (*First In First Out*), що завжди містить не більше 512 останніх вхідних ІКМ-аудіовідліків і на самому початку є заповненою нулями:

$$\mathbf{i} := 0..511, \quad \mathbf{FIFO_i} := 0.$$

[XXI] Надалі при оперуванні з елементами масивів нижня границя індексації останніх встановлена на рівні 0 (**ORIGIN** := 0).

Основні кроки алгоритму дій БФА наступні:

1) 32 нових відліки зі входу фільтра заносяться до буфера **X**, старі дані якого (на самому початку це нулі) одночасно зсуваються: нові відліки займають позиції буфера з індексами від 0 до 31 (найновіший відлік має індекс 0), а найстарші 32 відліки з буферу виключаються;
2) здійснюється поелементний добуток вектора буфера **X** і заданого стандартом вектора коефіцієнтів аналізуючого вікна ζ (рис. 5.111, табл. Г.1 Додатка)

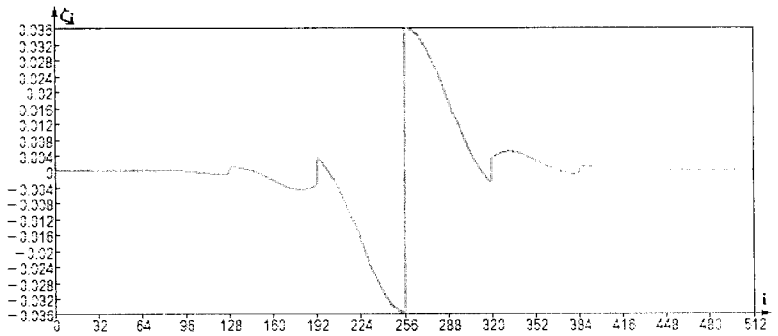


Рис. 5.111. Елементи стандартного аналізуючого вікна

у результаті чого одержують допоміжний 512-елементний вектор **Z**;

3) виконується проміжне обчислення 64-х значень Y_m ($m := 0..63$), для чого підсумовуються вісім елементів зі складу вектора **Z** ($j := 0..7$), що відстоять один від одного на 64 позиції ($m + 64 \cdot j$);
4) множенням елементів вектора **Y** на відповідні коефіцієнти матриці ДКП БФА **M** (рис. 5.112) обчислюються 32 субсмугових відліки V_s ($s := 0..31$):

$$M_{s,m} := \cos[(2 \cdot s + 1) \cdot (m - 16) \cdot \pi / 64];$$

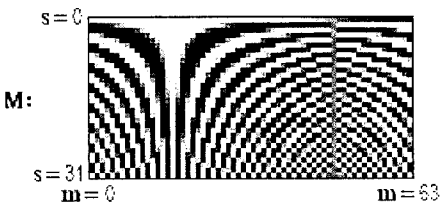


Рис. 5.112. Візуалізація матриці **M**

5) вихідні відліки БФА нормуються шляхом ділення на максимум абсолютного значення ІКМ-сигналу ($2^{\Delta A-1} = 32768$).

Програмний модуль субсмугового аналізу — (М.123)[XXII].

$$
\begin{aligned}
V := \;& F \leftarrow \text{FIFO} \\
& \text{for } a \in 0..N_A - 1 \\
& \quad T \leftarrow \text{subvec}[PCM, a \cdot S_K, (a+1) \cdot S_K - 1] \\
& \quad \text{for } b \in 0..35 \\
& \quad\quad O \leftarrow \text{reverse}[\text{subvec}[T, b \cdot 32, (b+1) \cdot 32 - 1]] \\
& \quad\quad F \leftarrow \text{subvec}(\text{stack}(O, F), 0, 511) \\
& \quad\quad Z \leftarrow \overrightarrow{(\zeta \cdot F)} \\
& \quad\quad \text{for } m \in 0..63 \\
& \quad\quad\quad \Sigma Y \leftarrow 0 \\
& \quad\quad\quad \text{for } j \in 0..7 \\
& \quad\quad\quad\quad \Sigma Y \leftarrow \Sigma Y + Z_{m+64 \cdot j} \\
& \quad\quad\quad Y_m \leftarrow \Sigma Y \\
& \quad\quad \text{for } s \in 0..31 \\
& \quad\quad\quad \Sigma V \leftarrow 0 \\
& \quad\quad\quad \text{for } m \in 0..63 \\
& \quad\quad\quad\quad \Sigma V \leftarrow \Sigma V + M_{s,m} \cdot Y_m \\
& \quad\quad\quad V_s \leftarrow \Sigma V \\
& \quad\quad V\_full_{b,a} \leftarrow V \\
& V\_full \\
& 32768
\end{aligned}
\qquad \text{(М.123)}
$$

Результатом виконання (М.123) є тривимірний масив **V**, індекси рядків якого відповідають індексам 36 субсмугових відліків (**b** := 0..35), індекси стовпців — індексам N_A аудіокадрів (**a** := 0.. N_A–1), а кожен елемент — це вектор **b**-х відліків на виході БФА у кожній з 32-х субсмуг (**s** := 0..31).

Крок 3

У кожній з 32-х субсмуг окремого аудіокадру для кожної з трьох послідовних груп по 12 субсмугових відліків ставиться у відповідність свій масштабний коефіцієнт (МК, **ScF** – *Scale Factor*) — загалом 3·32 = 96 МК — максимальне по модулю серед 12-ти відліків значення квантується за стандартною таблицею (табл. А.1 Додатка).

[XXII] Для компактності запису тут і у подальших програмних модулях цього підрозділу замість вбудованої функції *Mathcad* **submatrix**(...) використовується користувальницька функція **subvec**(**V**, **r1**, **r2**) := **submatrix**(**V**, **r1**, **r2**, 0, 0).

Програмний модуль обчислення МК для всіх N_A кадрів — (М.124).

```
MK := | ScF ← READFILE("SCF.xls" , "Excel")                    (М.124)
      | for a ∈ 0.. N_A − 1
      |   for s ∈ 0.. 31
      |     for b ∈ 0.. 35
      |       g1_b ← (V_{b,a})_s   if b < 12
      |       g2_{b−12} ← (V_{b,a})_s   if 11 < b < 24
      |       g3_{b−24} ← (V_{b,a})_s   if 23 < b
      |     max1 ← max(max(g1) , |min(g1)|)
      |     max2 ← max(max(g2) , |min(g2)|)
      |     max3 ← max(max(g3) , |min(g3)|)
      |     G1_s ← ScF_0   if max1 ≥ ScF_1
      |     G2_s ← ScF_0   if max2 ≥ ScF_1
      |     G3_s ← ScF_0   if max3 ≥ ScF_1
      |     if max1 < ScF_1
      |       j ← 0
      |       while j < 64 ∧ max1 > ScF_{63−j}
      |             j ← j + 1
      |       G1_s ← ScF_{63−j}
      |     if max2 < ScF_1
      |       j ← 0
      |       while j < 64 ∧ max2 > ScF_{63−j}
      |             j ← j + 1
      |       G2_s ← ScF_{63−j}
      |     if max3 < ScF_1
      |       j ← 0
      |       while j < 64 ∧ max3 > ScF_{63−j}
      |             j ← j + 1
      |       G3_s ← ScF_{63−j}
      |   MK_{0,a} ← G1 , MK_{1,a} ← G2 , MK_{2,a} ← G3
      | MK
```

Результатом виконання (М.124) є тривимірний масив **MK**, індекси рядків якого відповідають індексам 3-х груп субсмугових відліків (**g** := 0..2), індекси стовпців — індексам N_A аудіокадрів (**a** := 0.. N_A−1), а

кожен елемент — це вектор масштабних коефіцієнтів для 32-х субсмуг (**s** := 0..31). Приклад обчислення МК наведений на рис. 5.113.

$MK_{3,a}$ =

0	0.00048828125000
1	0.00310039267963
2	0.00390625000000
3	0.00310039267963
4	0.00077509816991
5	0.00246078330058
6	0.00030759791257
7	0.00030759791257
8	0.00012207031250
9	0.00015379895629
10	0.00019377454248
11	0.00038754908495
12	0.00007689947814
13	0.00007689947814
14	0.00006103515625
15	0.00015379895629
16	0.00007689947814
17	0.00001525878906
18	0.00001211090890
19	0.00004844363562
20	0.00001922486954
21	0.00001525878906
22	0.00000762939453
23	0.00000762939453
24	0.00000762939453
25	0.00000381469727
26	0.00000480621738
27	0.00000605545445
28	0.00000605545445
29	0.00000605545445
30	0.00000762939453
31	0.00000605545445

$MK_{1,a}$ =

0	0.00390625000000
1	0.01240157071850
2	0.00492156660115
3	0.00620078535925
4	0.00123039165029
5	0.00781250000000
6	0.00097656250000
7	0.00061519582514
8	0.00030759791257
9	0.00030759791257
10	0.00155019633981
11	0.00155019633981
12	0.00019377454248
13	0.00019377454248
14	0.00007689947814
15	0.00006103515625
16	0.00012207031250
17	0.00003051757813
18	0.00001922486954
19	0.00004844363562
20	0.00004844363562
21	0.00003051757813
22	0.00001211090890
23	0.00000961243477
24	0.00001922486954
25	0.00000961243477
26	0.00000762939453
27	0.00000961243477
28	0.00000961243477
29	0.00000605545445
30	0.00000762939453
31	0.00000480621738

$MK_{2,a}$ =

0	0.12500000000000
1	0.07874506561843
2	0.00492156660115
3	0.02480314143700
4	0.01562500000000
5	0.00781250000000
6	0.00492156660115
7	0.00048828125000
8	0.00195312500000
9	0.00310039267963
10	0.00246078330058
11	0.00246078330058
12	0.00123039165029
13	0.00012207031250
14	0.00097656250000
15	0.00123039165029
16	0.00097656250000
17	0.00061519582514
18	0.00009688727124
19	0.00015379895629
20	0.00015379895629
21	0.00030759791257
22	0.00009688727124
23	0.00006103515625
24	0.00012207031250
25	0.00006103515625
26	0.00003844973907
27	0.00001525878906
28	0.00000961243477
29	0.00000762939453
30	0.00000381469727
31	0.00000480621738

Рис. 5.113. Приклад результату обчислення масштабних коефіцієнтів для першого (**a** := 0) аудіокадру

Крок 4

Формування психоакустичної моделі (ПАМ) обробки вхідного ІКМ-аудіосигналу, що має кінцевим результатом визначення для кожної з

32-х субсмуг сигналу відношення рівня сигналу до порогу маскування *SMR*, враховуючи властивості ССЛ (див. Додаток Б, а також розділ А.1 Додатка). Програмні модулі реалізації ПАМ № 1 побудовані згідно Додатка Б.

1) Виконання ШПФ сегмента **x** вхідного ІКМ-сигналу, що відповідає окремому аудіокадру (див. розділ Б.1 Додатка), покладене на програмний модуль (М.125), що використовує вектор **h** вікна Ганна (Б.1) та вбудовану функцію прямого ШПФ (Б.2) у *Mathcad* — **FFT(h·x)**. Результатом виконання (М.125) є масив спектрального розподілу потужності **X**, стовпці якого відповідають N_A аудіокадрам, а рядки — $k := 0..N/2$ частотним відлікам кожного кадру вхідного сигналу.

Візуалізоване представлення результату наведене на рис. 5.114.

$$\begin{aligned}
&X := \left|\begin{array}{l}
\text{for } a \in 0..N_A - 1 \\
\quad x \leftarrow \text{subvec}\left[PCM, a\cdot S_K + \Delta, (a+1)\cdot S_K - \Delta - 1\right] \\
\quad \text{for } i \in 0..N-1 \\
\quad\quad h\cdot x_i \leftarrow h_i \cdot x_i \\
\quad Xa \leftarrow 20\cdot\log\left(\left|\overrightarrow{FFT(h\cdot x)}\right|\right) \\
\quad X^{(a)} \leftarrow Xa - \max(Xa) + 96.3296 \\
X
\end{array}\right. \qquad \text{(М.125)}
\end{aligned}$$

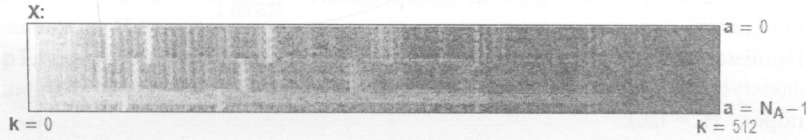


Рис. 5.114. Візуалізація транспонованого масиву спектрального розподілу потужності **X** (світлі ділянки відповідають більшій потужності; максимально — 96.33 дБ)

2) Визначення рівня звукового тиску для кожної субсмуги здійснюється у відповідності до розділу Б.2 Додатка за наступних вихідних умов:

- загальна кількість субсмуг розкладу ШПФ: $S_{cc} := 32$;
- ширина окремої субсмуги: $\Delta N := N/(2\cdot S_{cc}) = 1024/64 = 16$ відліків.

Результатом виконання програмного модуля обчислення розподілу енергії сигналу по субсмугах — (М.126) — є масив L_{cc}, кількість стовпців якого дорівнює кількості аудіокадрів N_A, а кількість рядків — кількості субсмуг S_{cc}. Приклад візуалізованого представлення масиву L_{cc} наведений на рис. 5.115.

$$
\begin{aligned}
&L_{cc} := \left|\begin{array}{l}
\text{for } a \in 0..N_A - 1 \\
\quad G \leftarrow \text{augment}\left(MK_{2,a}, MK_{1,a}, MK_{0,a}\right)^T \\
\quad \text{for } s \in 0..S_{cc} - 1 \\
\quad\quad MK_{max_{s,a}} \leftarrow \max\left(G^{\langle s \rangle}\right) \\
\quad\quad X_{лок.max} \leftarrow \max\left[\text{subvec}\left[X^{\langle a \rangle}, s \cdot \Delta N, (s+1) \cdot \Delta N - 1\right]\right] \\
\quad\quad L_{cc_{s,a}} \leftarrow \max\left(X_{лок.max}, 20 \cdot \log\left(MK_{max_{s,a}}\right) + 80.309\right) \\
L_{cc}
\end{array}\right. \qquad \text{(M.126)}
\end{aligned}
$$

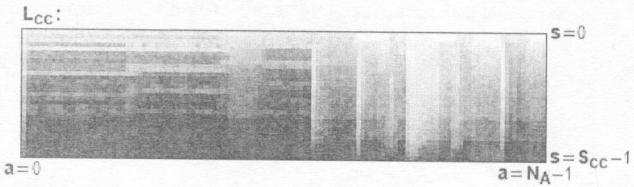


Рис. 5.115. Візуалізація масиву рівнів звукового тиску L_{cc} (світліші ділянки відповідають більш високому рівню потужності; максимально — 96.33 дБ)

3) Визначення абсолютного порогу (порогу у тиші) згідно розділу Б.3 Додатка здійснюється за попереднього задання загальної бітової швидкості передавання стиснутих зі втратами аудіоданих **R** на канал. Приймемо, що $R := 64 \cdot 10^3$ біт/с. Тоді вектор значень порогу у тиші **LTq** формується згідно виразу (Б.4) і даних табл. Б.1 (кількість значень порогу $H := 126$):

$$F\_CBR\_ATh := \text{READFILE}(\text{"Fr\_CBR\_ATh.xls"}, \text{"Excel"});$$

$$LTq := \text{if}\left(R \geq 96 \cdot 10^3, F\_CBR\_ATh^{\langle 3 \rangle} + \text{offset}, F\_CBR\_ATh^{\langle 3 \rangle}\right).$$

Оскільки при заданій швидкості **R** зсув $\text{offset} := \text{if}\left(R \geq 96 \cdot 10^3, -12, 0\right) = 0$ дБ, то одержана залежність порогу тиші від частоти відповідає наведеній на рис. Б.2 Додатка, а залежність елементів вектора порогу тиші від своїх індексів — наведеній на рис. 5.116.

Відображення частот відліків вхідного сигналу $F_i := F\_CBR\_ATh^{\langle 1 \rangle}$ ($i := 0..H-1$; $H = 126$) на коефіцієнти (індекси) з таблиці абсолютних порогів (див. табл. Б.1. Додатка) здійснюється за допомогою програмного модуля (М.127).

Результуючий вектор **Map** містить $N/2+1$ елементів. Зв'язок елементів вектора відображення **Map** з частотами відліків вхідного сигналу ілюструє рис. 5.117.

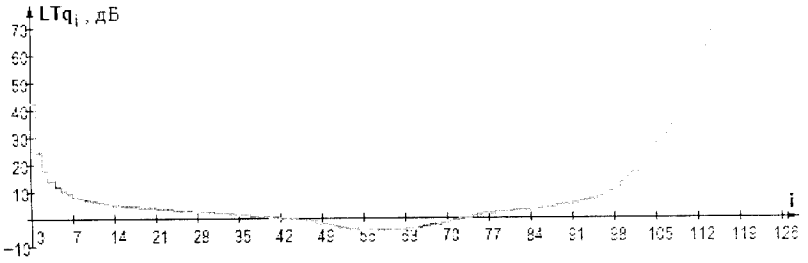


Рис. 5.116. Графік вмісту елементів вектора **LTq**

$$\textbf{Map} := \left| \begin{array}{l} \textbf{for}\ \ i \in 0..\ H-1 \\ \quad \textbf{Fr\_index}_i \leftarrow \textbf{round}\left[\dfrac{F_i}{f_д}\cdot N\right] \\ \textbf{Map}_0 \leftarrow \textbf{Fr\_index}_0 \\ \textbf{for}\ \ i \in 2..\ H-1 \\ \quad \textbf{for}\ \ k \in \textbf{Fr\_index}_{i-1}..\ \textbf{Fr\_index}_i - 1 \\ \qquad \textbf{Map}_{k-1} \leftarrow i \\ \textbf{for}\ \ k \in \textbf{Fr\_index}_{H-1}..\ \dfrac{N}{2}+1 \\ \quad \textbf{Map}_{k-1} \leftarrow H \\ \textbf{Map} \end{array} \right. \qquad \text{(M.127)}$$

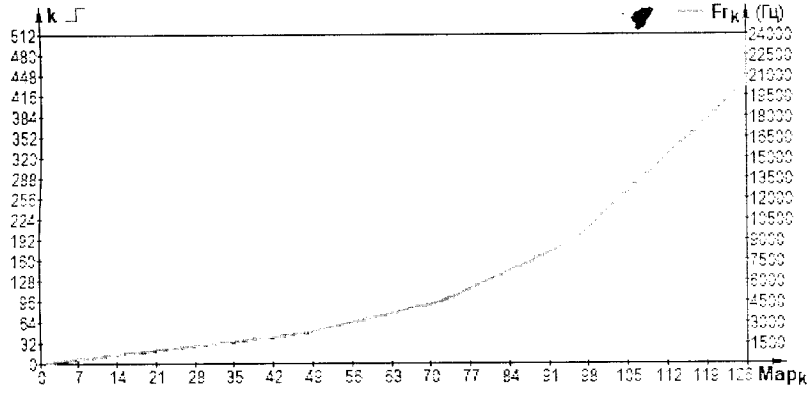


Рис. 5.117. Графік залежності елементів вектора відображення **Map** від своїх індексів та частот відліків

4) Пошук тональних (ТК) і нетональних (НТК) компонент починається з маркування локальних максимумів у відповідності до умов (Б.5) (див. підрозділ Б.4.1 Додатка), що реалізується програмним модулем (М.128).

$$\begin{pmatrix} K_{лм} \\ X_{лм} \end{pmatrix} := \left| \begin{array}{l} \text{for } a \in 0..N_A - 1 \\ \quad \left| \begin{array}{l} c \leftarrow 0 \\ \text{for } k \in 1..\frac{N}{2} - 2 \\ \quad \text{if } \left(X_{k-1,a} < X_{k,a}\right) \wedge \left(X_{k+1,a} \le X_{k,a}\right) \\ \quad\quad \left| \begin{array}{l} K_{лм_{c,a}} \leftarrow k,\ X_{лм_{c,a}} \leftarrow X_{k,a} \\ c \leftarrow c + 1 \end{array} \right. \end{array} \right. \\ \begin{pmatrix} K_{лм} \\ X_{лм} \end{pmatrix} \end{array} \right. \qquad \text{(М.128)}$$

Результатом виконання (М.128) є масиви порядкових індексів ($K_{лм}$) і потужностей ($X_{лм}$) локальних максимумів для кожного аудіокадру. Результат пошуку локальних максимумів для першого ($a = 0$) аудіокадру наведено на рис. 5.118. Взаємозалежне відтворення усіх елементів $X_{лм}$ і $K_{лм}$ представлене на рис. 5.119.

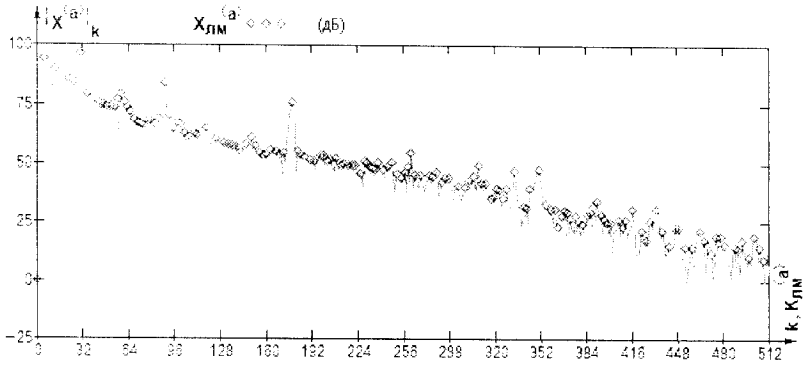


Рис. 5.118. Приклад результату пошуку локальних максимумів у спектрі аудіокадру $a = 0$

Програмний модуль пошуку ТК — (М.129) — повертає масиви порядкових номерів ($K_{тк}$) і рівнів звукового тиску ($X_{тк}$) тональних компонент, а також масив **FLAG** міток тональних і відкинутих компонент. Кількість стовпців у зазначених масивах відповідає кількості аудіокадрів N_A. Кіль-

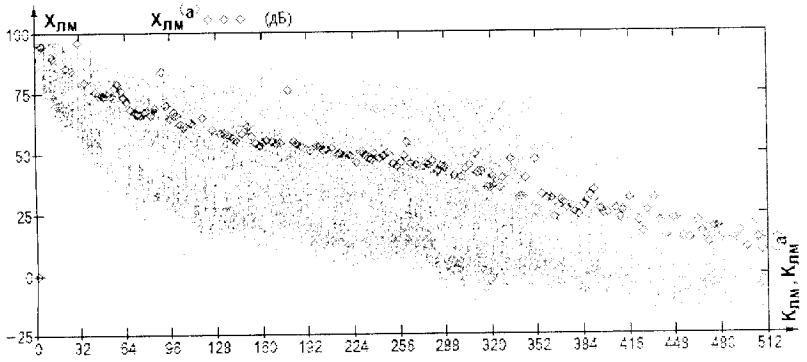


Рис. 5.119. Взаємозалежне відображення елементів масивів $\mathbf{X_{лм}}$ і $\mathbf{K_{лм}}$

кість же рядків у масивах $\mathbf{K_{тк}}$ і $\mathbf{X_{тк}}$ залежить від максимальної кількості компонент, які було охарактеризовано як тональні. Кількість рядків у масиві **FLAG** дорівнює кількості частотних відліків ШПФ — $\mathbf{N}/2+1$.

$$
\begin{pmatrix} K_{тк} \\ X_{тк} \\ FLAG \end{pmatrix} :=
$$

```
J1 ← (−2 2)^T,  J2 ← (−3 −2 2 3)^T,  J3 ← (−6 −5 −4 −3 −2 2 3 4 5 6)^T
J4 ← (−12 −11 −10 −9 −8 −7 −6 −5 −4 −3 −2 2 3 4 5 6 7 8 9 10 11 12)^T
K_тк ← (0 1),  X_тк ← (0 1),  FLAG ← 0 "X"
for a ∈ 0.. N_A − 1
    Клм ← K_лм^⟨a⟩,  zero ← rows(пошук(0, Клм)),  c ← 0
    if rows(Клм) ≠ zero
        for i ∈ 0.. last(Клм) − zero
            k ← Клм_i,  is_tonal ← if(2 < k ≤ 500, 1, 0),  J ← (0)
            J ← if(2 < k < 63, J1, if(63 ≤ k < 127, J2, if(127 ≤ k < 255, J3, J4)))
            for j ∈ 0.. last(J)    if 2 < k ≤ 500
                is_tonal ← is_tonal ∧ (X_{k,a} − X_{k+J_j,a} ≥ 7)
            if is_tonal = 1
                K_тк_{c,a} ← k
                X_тк_{c,a} ← 10·log(10^(0.1·X_{k−1,a}) + 10^(0.1·X_{k,a}) + 10^(0.1·X_{k+1,a}))
                FLAG_{k,a} ← "TK",  J' ← stack[J, (−1 1)^T]
                for j ∈ 0.. last(J')
                    FLAG_{k+J'_j,a} ← "Відкинуто"
                c ← c + 1
(K_тк)
 X_тк
 FLAG)
```

(M.129)

Результат пошуку тональних компонент для першого (**a** = 0) аудіокадру наведено на рис. 5.120. Взаємозалежне відтворення усіх елементів $X_{тк}$ і $K_{тк}$ — на рис. 5.121. Візуалізоване представлення результату формування списку тональних і відкинутих компонент показане на рис. 5.122.

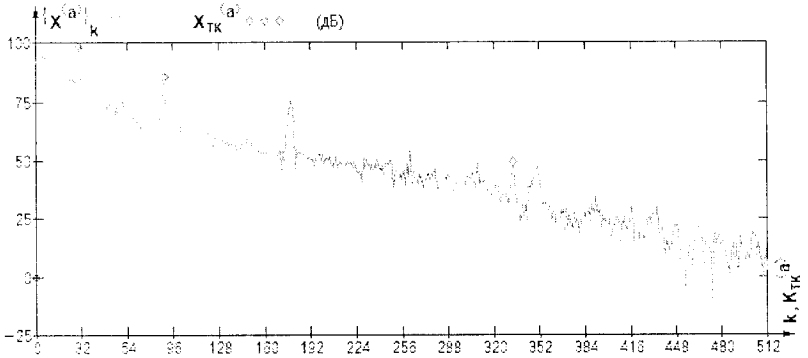


Рис. 5.120. Приклад результату пошуку тональних компонент у спектрі аудіокадру **a** = 0

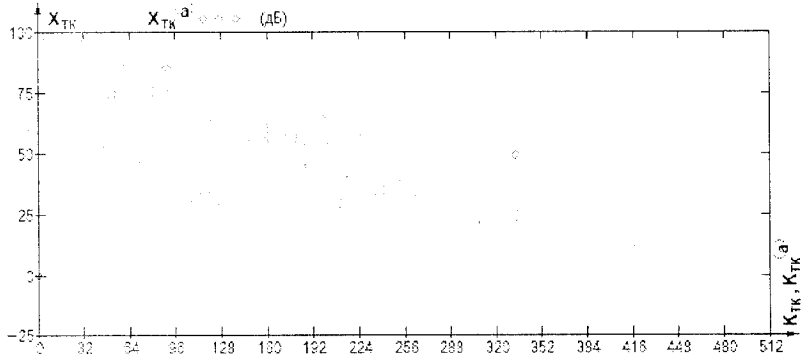


Рис. 5.121. Взаємозалежне відображення елементів масивів $X_{тк}$ і $K_{тк}$

Рис. 5.122. Візуалізація транспонованого масиву спектра потужності **X** з тональними (білі) і відкинутими (чорні) елементами

Процедура формування списку нетональних компонент і обчислення їх потужності (див. підрозділ Б.4.3 Додатка) виконується програмним модулем (М.130), що повертає масиви порядкових номерів ($\mathbf{K}_{\text{нтк}}$) і потужностей ($\mathbf{X}_{\text{нтк}}$) нетональних компонент, а також оновлений масив **FLAG** міток тональних, нетональних і відкинутих компонент.

$$\begin{pmatrix} \mathbf{K}_{\text{нтк}} \\ \mathbf{X}_{\text{нтк}} \\ \mathbf{FLAG} \end{pmatrix} := \left| \begin{array}{l} \mathbf{CBB} \leftarrow \mathbf{READFILE}(\text{"CBB.xls"}, \text{"Excel"}) \\ \mathbf{K}_{\text{нтк}} \leftarrow 101,\ \mathbf{X}_{\text{нтк}} \leftarrow 101,\ \mathbf{FLAG} \leftarrow \mathbf{FLAG},\ P_{min} \leftarrow -200\ \text{дБ} \\ \textbf{for}\ \ a \in 0..\ N_A - 1 \\ \quad \text{№} \leftarrow \mathbf{CBB}^{\langle 1 \rangle} \\ \quad \textbf{for}\ \ i \in 0..\ \mathbf{rows}(\mathbf{CBB}) - 2 \\ \quad\quad \mathbf{power} \leftarrow P_{min},\ \mathbf{weight} \leftarrow 0 \\ \quad\quad \textbf{for}\ \ k \in \mathbf{index}_{\text{№}_{i-1}} - 1\ ..\ \mathbf{index}_{\text{№}_{i+1}} - 2 \\ \quad\quad\quad \textbf{if}\ \ \mathbf{FLAG}_{k,a} = 0 \\ \quad\quad\quad\quad \mathbf{power} \leftarrow 10 \cdot \log\left(10^{0.1 \cdot \mathbf{power}} + 10^{0.1 \cdot \mathbf{X}_{k,a}}\right) \\ \quad\quad\quad\quad \mathbf{weight} \leftarrow \mathbf{weight} + 10^{0.1 \cdot \mathbf{X}_{k,a}} \cdot \left[\mathbf{CBR}_{(\mathbf{Map}_k)} - \mathbf{CBR}_{\text{№}_{i-1}}\right] \\ \quad\quad\quad\quad \mathbf{FLAG}_{k,a} \leftarrow \text{'Відкинуто'} \\ \quad\quad \textbf{if}\ \ \mathbf{power} > P_{min} \\ \quad\quad\quad K \leftarrow \mathbf{index}_{\text{№}_{i-1}} + \mathbf{round}\left[\dfrac{\mathbf{weight}}{10^{0.1 \cdot \mathbf{power}}} \cdot \left(\mathbf{index}_{\text{№}_{i+1}} - 1 - \mathbf{index}_{\text{№}_{i-1}}\right)\right] - 1 \\ \quad\quad\quad K \leftarrow \mathbf{rows}(\mathbf{FLAG}) - 1 \ \ \textbf{if}\ \ K > \mathbf{rows}(\mathbf{FLAG}) - 1 \\ \quad\quad\quad K \leftarrow K + 1 \ \ \textbf{if}\ \ \mathbf{FLAG}_{K,a} = \text{'ТК'} \\ \quad\quad\quad \mathbf{K}_{\text{нтк}_{i,a}} \leftarrow K,\ \mathbf{X}_{\text{нтк}_{i,a}} \leftarrow \mathbf{power},\ \mathbf{FLAG}_{K,a} \leftarrow \text{'НТК'} \\ \begin{pmatrix} \mathbf{K}_{\text{нтк}} \\ \mathbf{X}_{\text{нтк}} \\ \mathbf{FLAG} \end{pmatrix} \end{array} \right. \qquad \text{(М.130)}$$

Результат пошуку НТК для першого ($\mathbf{a} = 0$) аудіокадру і взаємозалежне відтворення усіх елементів масивів $\mathbf{X}_{\text{нтк}}$ і $\mathbf{K}_{\text{нтк}}$ показані на рис. 5.123, 5.124.

Візуалізація результуючого розміщення тональних, нетональних і відкинутих компонент (позначені чорними, сірими і білими крапками відповідно) представлена на рис. 5.125.

5) Прорідження (децимація) одержаних у попередньому пункті списків ТК і НТК здійснюється у відповідності до розділу Б.5 Додатка за допомогою програмних модулів (М.131), (М.132) і (М.133).

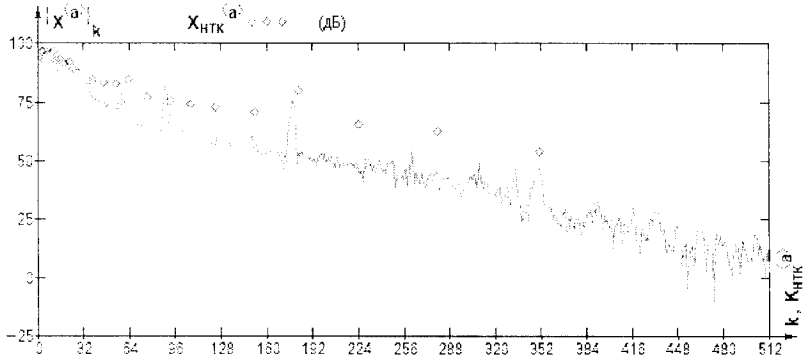


Рис. 5.123. Приклад результату пошуку нетональних компонент у спектрі аудіокадру **a** = 0

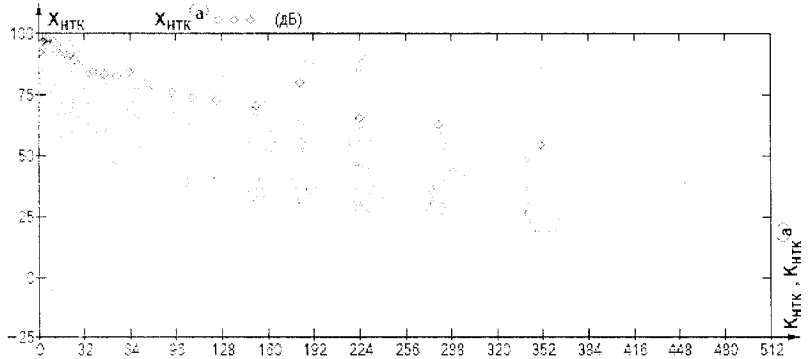


Рис. 5.124. Взаємозалежне відображення елементів масивів $\mathbf{X}_{\text{нтк}}$ і $\mathbf{K}_{\text{нтк}}$

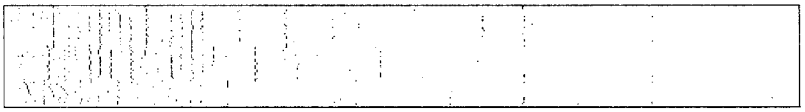


Рис. 5.125. Візуалізація розміщення елементів тональних (чорні), нетональних (сірі) і відкинутих (білі) компонент у спектрі ШНФ кожного з $\mathbf{N}_{\mathbf{A}}$ аудіокадрів

Три представлені нижче програмні модулі повертають проріджені масиви порядкових номерів ($\mathbf{dK}_{\text{нтк}}$ і $\mathbf{dK}_{\text{тк}}$) і потужностей ($\mathbf{dX}_{\text{нтк}}$ і $\mathbf{dX}_{\text{тк}}$) НТК і ТК, а також оновлений масив **dFLAG** міток проріджених компонент.

$$\begin{pmatrix} dK_{НТК} \\ dX_{НТК} \\ dFLAG \end{pmatrix} := \left| \begin{array}{l} dFLAG \leftarrow FLAG \ , \ dX_{НТК} \leftarrow \{0\} \\ \text{for } a \in 0..N_A - 1 \\ \quad c \leftarrow 0 \\ \quad \text{for } i \in 0..rows(K_{НТК}) - 1 \\ \quad\quad k \leftarrow K_{НТК_{i,a}} \\ \quad\quad dFLAG_{k,a} \leftarrow \text{'Відкинуто'} \ \text{ if } \ X_{НТК_{i,a}} < LTq_{(Map_k - 1)} \\ \quad\quad \text{if } \ X_{НТК_{i,a}} \geq LTq_{(Map_k - 1)} \\ \quad\quad\quad dK_{НТК_{c,a}} \leftarrow k \ , \ dX_{НТК_{c,a}} \leftarrow X_{НТК_{i,a}} \\ \quad\quad\quad c \leftarrow c + 1 \\ \begin{pmatrix} dK_{НТК} \\ dX_{НТК} \\ dFLAG \end{pmatrix} \end{array} \right. \qquad \text{(M.131)}$$

$$\begin{pmatrix} dK_{ТК1} \\ dX_{ТК1} \\ dFLAG \end{pmatrix} := \left| \begin{array}{l} dFLAG \leftarrow FLAG \ , \ dX_{ТК1} \leftarrow \{0\} \\ \text{for } a \in 0..N_A - 1 \\ \quad c \leftarrow 0 \\ \quad \text{for } i \in 0..rows(K_{ТК}) - 1 \\ \quad\quad k \leftarrow K_{ТК_{i,a}} \\ \quad\quad dFLAG_{k,a} \leftarrow \text{'Відкинуто'} \ \text{ if } \ X_{ТК_{i,a}} < LTq_{(Map_k - 1)} \\ \quad\quad \text{if } \ X_{ТК_{i,a}} \geq LTq_{(Map_k - 1)} \\ \quad\quad\quad dK_{ТК1_{c,a}} \leftarrow k \ , \ dX_{ТК1_{c,a}} \leftarrow X_{ТК_{i,a}} \\ \quad\quad\quad c \leftarrow c + 1 \\ \begin{pmatrix} dK_{ТК1} \\ dX_{ТК1} \\ dFLAG \end{pmatrix} \end{array} \right. \qquad \text{(M.132)}$$

У розгляданому випадку до прорідження загальна кількість ТК становила: **rows(пошук** ("ТК", **FLAG**)) = 1021. Після першого прорідження у (M.132) кількість ТК скоротилася до **rows(пошук** ("ТК", **dFLAG**)) = 967. Після додаткового прорідження у (M.133) остаточна кількість ТК стала становити **rows(пошук** ("ТК", **dFLAG**)) = 871.

Кількість НТК до прорідження: **rows(пошук** ("НТК", **FLAG**)) = 3059. Після прорідження у (M.131) кількість НТК: **rows(пошук** ("НТК", **dFLAG**)) = 2794.

Візуалізація підсумкового після прорідження розміщення тональних, нетональних і відкинутих компонент представлена на рис. 5.126.

6) Індивідуальні пороги маскування ТК і НТК (див. розділ Б.6 Додатка) обчислюються програмними модулями (M.134) і (M.135).

$$
\begin{pmatrix} dK_{TK} \\ dX_{TK} \\ dFLAG \end{pmatrix} :=
\left|
\begin{array}{l}
dK_{TK2} \leftarrow dK_{TK1}\ ,\ dX_{TK2} \leftarrow dX_{TK1}\ ,\ dFLAG \leftarrow dFLAG \\
\text{for}\ \ a \in 0..\ N_A - 1 \\
\quad i \leftarrow 0 \\
\quad N_{TK} \leftarrow \text{rows}\left(\text{пошук}\left('TK'\ , dFLAG^{\langle a \rangle}\right)\right) \\
\quad \text{while}\ \ i < N_{TK} - 1 \\
\quad\quad k1 \leftarrow dK_{TK2_{i,a}}\ ,\ \ k2 \leftarrow dK_{TK2_{i+1,a}} \\
\quad\quad \text{if}\ \ CBR_{\left(Map_{k2}-1\right)} - CBR_{\left(Map_{k1}-1\right)} < 0.5 \\
\quad\quad\quad \text{if}\ \ dX_{TK2_{i,a}} < dX_{TK2_{i+1,a}} \\
\quad\quad\quad\quad dK_{TK2}^{\langle a \rangle} \leftarrow \text{remove}\left(dK_{TK2}^{\langle a \rangle}, i\right) \\
\quad\quad\quad\quad dX_{TK2}^{\langle a \rangle} \leftarrow \text{remove}\left(dX_{TK2}^{\langle a \rangle}, i\right) \\
\quad\quad\quad\quad dFLAG_{k1,a} \leftarrow \text{'Відкинуто'} \\
\quad\quad\quad \text{otherwise} \\
\quad\quad\quad\quad dK_{TK2}^{\langle a \rangle} \leftarrow \text{remove}\left(dK_{TK2}^{\langle a \rangle}, i+1\right) \\
\quad\quad\quad\quad dX_{TK2}^{\langle a \rangle} \leftarrow \text{remove}\left(dX_{TK2}^{\langle a \rangle}, i+1\right) \\
\quad\quad\quad\quad dFLAG_{k2,a} \leftarrow \text{'Відкинуто'} \\
\quad\quad\quad N_{TK} \leftarrow \text{rows}\left(\text{пошук}\left('TK'\ , dFLAG^{\langle a \rangle}\right)\right) \\
\quad\quad i \leftarrow i + 1 \quad \text{if}\ \ CBR_{\left(Map_{k2}-1\right)} - CBR_{\left(Map_{k1}-1\right)} \geq 0.5 \\
\begin{pmatrix} dK_{TK} \\ dX_{TK} \\ dFLAG \end{pmatrix}
\end{array}
\right.
\qquad \text{(M.133)}
$$

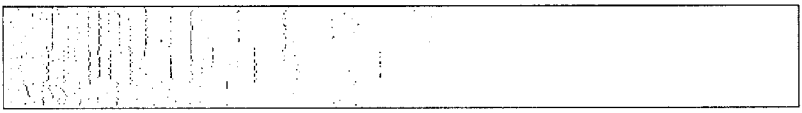


Рис. 5.126. Візуалізація розміщення елементів ТК (чорні), НТК (сірі) і відкинутих компонент (білі) після проведеної децимації

Результати обчислення індивідуальних порогів маскування ТК і НТК для першого (a = 0) аудіокадру наведено на рис. 5.127 і 5.128 відповідно.

$LT_{TK} :=$ | $p(k,u) \leftarrow -200$ дБ (М.134)
for $a \in 0..N_A - 1$
$N_{TK} \leftarrow rows(пошук("ТК", dFLAG^{(a)}))$
$LT_{TKa} \leftarrow if(N_{TK} > 0, matrix(N_{TK}, last(CBR), p), 0)$
for $u \in 0..last(CBR)$ if $N_{TK} > 0$
$z(u) \leftarrow CBR_u$
for $k \in 0..N_{TK} - 1$
$j \leftarrow dK_{TK_{k,a}}$, $Xj \leftarrow X_{j,a}$, $z(j) \leftarrow CBR_{(Map_j - 1)}$
$dz \leftarrow z(u) - z(j)$
if $-3 \le dz < 8$
$av_{TK} \leftarrow -6.025 - 0.275 \cdot z(j)$
$vf \leftarrow 17 \cdot dz - 0.4 \cdot Xj + 11$ if $-3 \le dz < -1$
$vf \leftarrow (0.4 \cdot Xj + 6) \cdot dz$ if $-1 \le dz < 0$
$vf \leftarrow -17 \cdot dz$ if $0 \le dz < 1$
$vf \leftarrow (1 - dz) \cdot (17 - 0.15 \cdot Xj) - 17$ if $1 \le dz < 8$
$LTка_{k,u} \leftarrow dX_{TK_{k,a}} + av_{TK} + vf$
$LT_{TK_a} \leftarrow LTка$
LT_{TK}

$LT_{НТК} :=$ | $p(k,u) \leftarrow -200$ дБ (М.135)
for $a \in 0..N_A - 1$
$N_{НТК} \leftarrow rows(пошук("НТК", dFLAG^{(a)}))$
$LTнтка \leftarrow if(N_{НТК} > 0, matrix(N_{НТК}, last(CBR), p), 0)$
for $u \in 0..last(CBR)$ if $N_{НТК} > 0$
$z(u) \leftarrow CBR_u$
for $k \in 0..N_{НТК} - 1$
$j \leftarrow dK_{НТК_{k,a}}$, $Xj \leftarrow X_{j,a}$, $z(j) \leftarrow CBR_{(Map_j - 1)}$
$dz \leftarrow z(u) - z(j)$
if $-3 \le dz < 8$
$av_{НТК} \leftarrow -2.025 - 0.175 \cdot z(j)$
$vf \leftarrow 17 \cdot dz - 0.4 \cdot Xj + 11$ if $-3 \le dz < -1$
$vf \leftarrow (0.4 \cdot Xj + 6) \cdot dz$ if $-1 \le dz < 0$
$vf \leftarrow -17 \cdot dz$ if $0 \le dz < 1$
$vf \leftarrow (1 - dz) \cdot (17 - 0.15 \cdot Xj) - 17$ if $1 \le dz < 8$
$LTнтка_{k,u} \leftarrow dX_{НТК_{k,a}} + av_{НТК} + vf$
$LT_{НТК_a} \leftarrow LTнтка$
$LT_{НТК}$

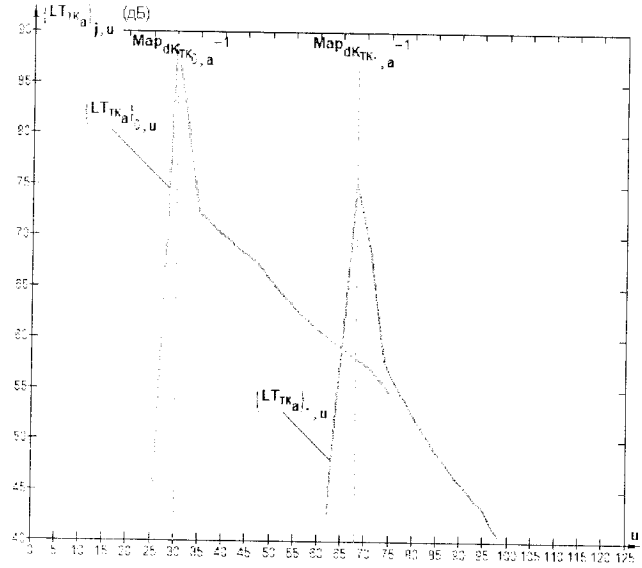


Рис. 5.127. Приклад результату обчислення індивідуальних порогів маскування ТК

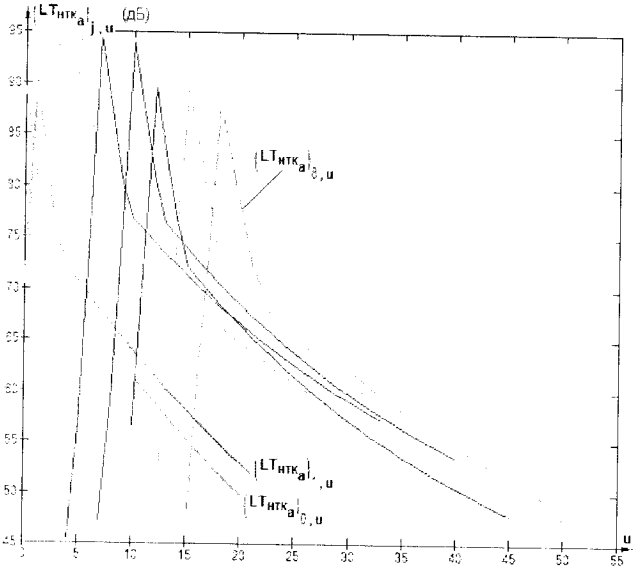


Рис. 5.128. Приклад результату обчислення індивідуальних порогів маскування НТК

7) Обчислення глобального порогу маскування (див. розділ Б.7 Додатка) здійснюється програмним модулем (М.136), у відповідності до виразу (Б.14).

```
LTg := | for a ∈ 0.. N_A - 1                                                                    (М.136)
       |   LTка/10 ← 0.1·LTк_a , LTнтка/10 ← 0.1·LTнтк_a , LTq/10 ← 0.1·LTq
       |   Nтк ← rows(LTка/10) , Nнтк ← rows(LTнтка/10)
       |   for u ∈ 0.. last(LTq)
       |     LTg_{u,a} ← 10·log[ 10^(LTq/10_u) + Σ_{j=0}^{Nтк-1} 10^(0 on error LTка/10_{j,u}) + Σ_{j=0}^{Nнтк-1} 10^(0 on error LTнтка/10_{j,u}) ]
       | LTg
```

Результат визначення глобального порогу маскування на **u**-х частотних відліках для першого аудіокадру (**a** = 0) показаний на рис. 5.129. Зв'язок елементів вектора відображення $\mathbf{LTg}^{<a>}$ з частотами відліків вхідного сигналу ілюструє рис. 5.130.

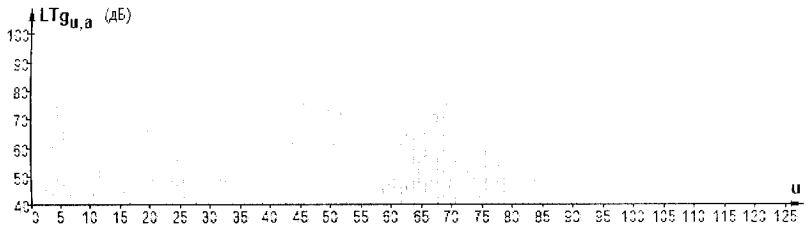


Рис. 5.129. Приклад результату обчислення глобального порогу маскування

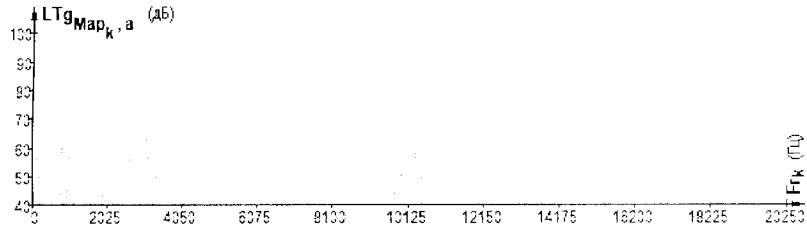


Рис. 5.130. Приклад залежності елементів вектора глобального порогу маскування від частот відліків

8) Мінімальні пороги маскування для всіх субсмуг кожного з $\mathbf{N_A}$ аудіокадрів визначаються відповідно до процедури (Б.15) розділу Б.8

Додатка, реалізацію якої покладено на програмний модуль (М.137). Результатом обчислення є масив **LTmin**, кількість рядків якого дорівнює кількості субсмуг S_{cc}, а кількість стовпців — кількості аудіокадрів N_A (рис. 5.131). Результат обчислення мінімальних порогів маскування для першого (**a** = 0) аудіокадру наведений на рис. 5.132.

$$\begin{aligned}&\mathbf{LTmin} := \left|\begin{array}{l}\text{for } a \in 0..N_A - 1\\ \quad \text{for } s \in 0..S_{CC} - 1\\ \quad\quad \mathbf{LTmin}_{s,a} \leftarrow \min\left(\mathbf{subvec}\left(\mathbf{LTg}^{(a)}, \mathbf{Map}_{s\cdot\Delta N} - 1, \mathbf{Map}_{s\cdot\Delta N+\Delta N-1} - 1\right)\right)\\ \mathbf{LTmin}\end{array}\right.\end{aligned} \qquad \text{(М.137)}$$

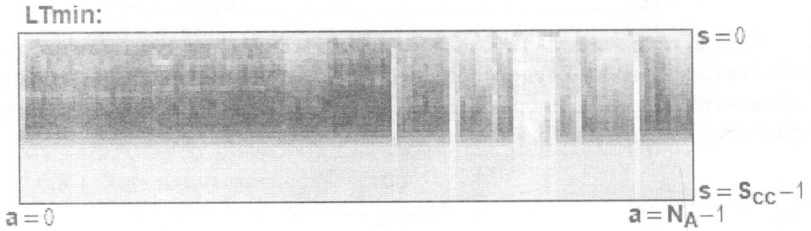


Рис. 5.131. Візуалізація масиву мінімальних порогів маскування **LTmin** (світліші ділянки відповідають більш високому рівню потужності)

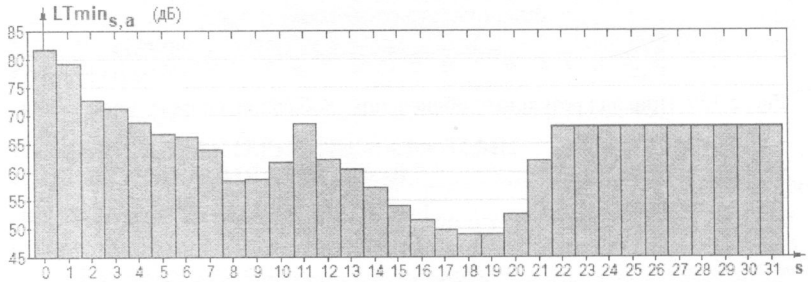


Рис. 5.132. Приклад результату визначення мінімальних порогів маскування

9) Відношення рівня сигналу до порогу маскування є результатом обчислення різниці двох співрозмірних масивів: рівнів звукового тиску L_{cc} та мінімальних порогів маскування **LTmin** — див. вираз (Б.16) розділу Б.9 Додатка, а також рис. 5.115 і 5.131. Візуалізоване представлення підсумкового масиву **SMR** наведене на рис. 5.133, а результат визначення відношення «сигнал/поріг маскування» для першого (**a** = 0) аудіокадру представлений на рис. 5.134.

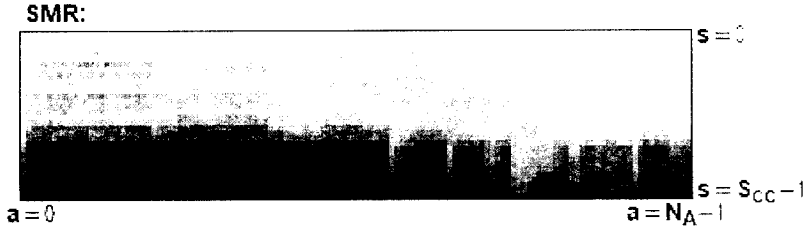


Рис. 5.133. Візуалізація масиву мінімальних порогів маскування **LTmin** (світліші ділянки відповідають більш високому рівню потужності)

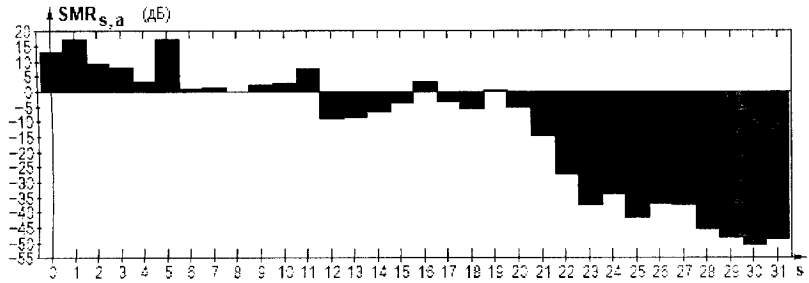


Рис. 5.134. Приклад результату визначення відношень «сигнал/поріг маскування»

Крок 5

Підсумкове кодування ІМК та ІВМК передбачає на своєму початку обчислення різниць індексів сусідніх (за групами) МК (див. розділ А.4 Додатка). Виконання такої процедури покладене на програмний модуль (М.138). Візуалізований результат обчислення масивів різниць $\mathbf{P_1}$ і $\mathbf{P_2}$, що мають кінцеву розмірність $\mathbf{S_{cc}} \times \mathbf{N_A}$, показаний на рис. 5.135.

У відповідності до значень різниць ІМК, керуючись табл. А.2 Додатка, програмний модуль (М.139) присвоює обчисленим різницям відповідний клас (від 1-го по 5-й), формуючи масиви $\mathbf{KP_1}$ і $\mathbf{KP_2}$ (* = 1 або 2), що мають ту ж саму розмірність ($\mathbf{S_{cc}} \times \mathbf{N_A}$). Візуалізацію результату наведено на рис. 5.136.

Пара елементів масивів класів різниць $\mathbf{KP_1}$ і $\mathbf{KP_2}$ з однаковими індексами однозначно вказує на конфігурацію передавання МК (див. табл. А.3 Додатка). Усі можливі конфігурації передавання ІМК і кодів ІВМК для зручності попередньо зводяться до відповідних масивів **Конф_пер** і **ТІВМК**, в яких номер рядка і номер стовпця (індексів елемента конфігурації або коду) визначаються значеннями елементів (класами) масивів $\mathbf{KP_1}$ і $\mathbf{KP_2}$ — (М.140).

$$\begin{pmatrix} P_1 \\ P_2 \end{pmatrix} := \left| \begin{array}{l} \text{for } a \in 0..N_A - 1 \\ \quad G1 \leftarrow MK_{0,a},\ G2 \leftarrow MK_{1,a},\ G3 \leftarrow MK_{2,a} \\ \quad \text{for } s \in 0..S_{CC} - 1 \\ \quad\quad «1» \leftarrow \text{пошук}(G1_s, ScF)_0 \\ \quad\quad «2» \leftarrow \text{пошук}(G2_s, ScF)_0 \\ \quad\quad «3» \leftarrow \text{пошук}(G3_s, ScF)_0 \\ \quad\quad P1_{s,a} \leftarrow \text{ІМК}_{«1»} - \text{ІМК}_{«2»} \\ \quad\quad P2_{s,a} \leftarrow \text{ІМК}_{«2»} - \text{ІМК}_{«3»} \\ \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} \end{array} \right. \qquad \text{(М.138)}$$

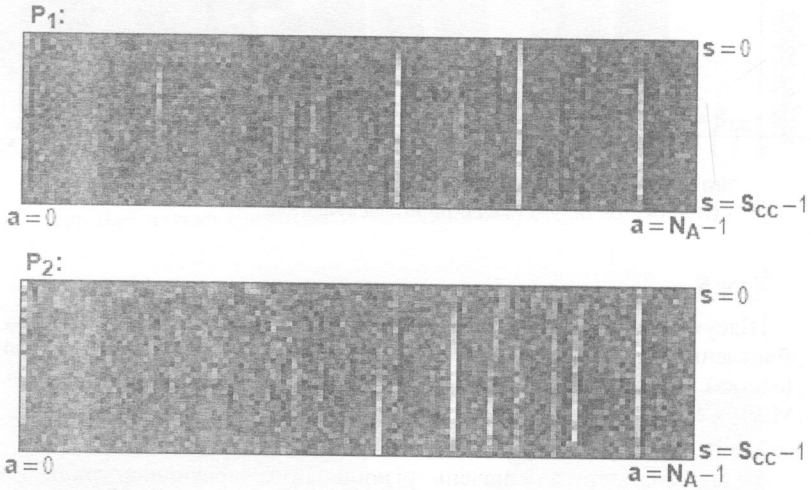


Рис. 5.135. Візуалізація масивів різниць ІМК P_1 і P_2 (світліші ділянки відповідають більш додатним різницям, темніші — більш від'ємним)

Програмний модуль формування масиву ІМК, що можуть бути передані з урахуванням конфігурації передавання й у разі обрання у подальшому (під час динамічного розподілу бітів) поточної субсмуги для кодування — (М.141), — результатом виконання якого є тривимірний масив **ІМКпер**, що має варіюючу розмірність: $S_{CC} \times N_A \times (1...3\ \text{ІМК})$.

$$KP_{\star} := \left| \begin{array}{l} \text{for } a \in 0..N_A - 1 \\ \quad \text{for } s \in 0..S_{CC} - 1 \\ \quad\quad \left(KP_{\star}\right)_{s,a} \leftarrow 1 \text{ if } \left(P_{\star}\right)_{s,a} \le -3 \\ \quad\quad \left(KP_{\star}\right)_{s,a} \leftarrow 2 \text{ if } -3 < \left(P_{\star}\right)_{s,a} < 0 \\ \quad\quad \left(KP_{\star}\right)_{s,a} \leftarrow 3 \text{ if } \left(P_{\star}\right)_{s,a} = 0 \\ \quad\quad \left(KP_{\star}\right)_{s,a} \leftarrow 4 \text{ if } 0 < \left(P_{\star}\right)_{s,a} < 3 \\ \quad\quad \left(KP_{\star}\right)_{s,a} \leftarrow 5 \text{ if } \left(P_{\star}\right)_{s,a} \ge 3 \\ KP_{\star} \end{array} \right. \qquad \text{(M.139)}$$

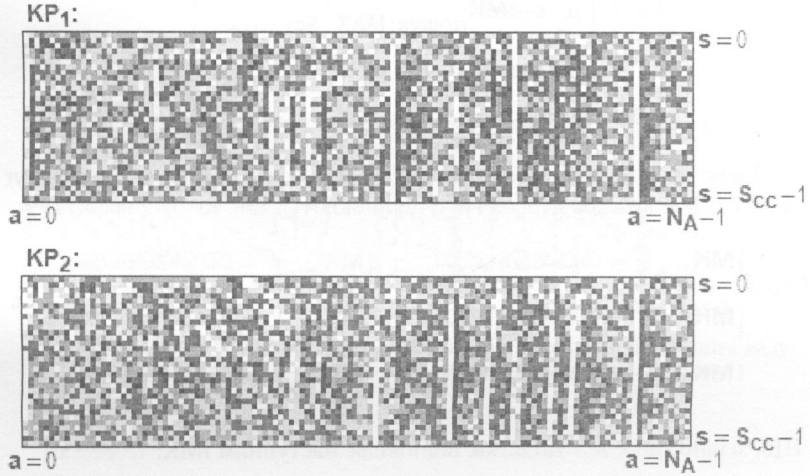


Рис. 5.136. Візуалізація масивів класів різниць ІМК KP_1 і KP_2 (білі ділянки відповідають класу «5», чорні — «1», у відтінках сірого — від «2» по «4»)

$$\text{Конф\_пер} := \begin{bmatrix} \begin{pmatrix}1\\2\\3\end{pmatrix} & \begin{pmatrix}1\\2\end{pmatrix} & \begin{pmatrix}1\\2\end{pmatrix} & \begin{pmatrix}1\\3\end{pmatrix} & \begin{pmatrix}1\\2\\3\end{pmatrix} \\ \begin{pmatrix}1\\3\end{pmatrix} & (1) & (1) & (4) & \begin{pmatrix}1\\3\end{pmatrix} \\ (1) & (1) & (1) & (3) & \begin{pmatrix}1\\3\end{pmatrix} \\ (2) & (2) & (2) & (3) & \begin{pmatrix}1\\2\\3\end{pmatrix} \\ \begin{pmatrix}1\\2\\3\end{pmatrix} & \begin{pmatrix}1\\2\end{pmatrix} & \begin{pmatrix}1\\2\end{pmatrix} & \begin{pmatrix}1\\3\end{pmatrix} & \begin{pmatrix}1\\2\\3\end{pmatrix} \end{bmatrix}; \quad \text{ТІВМК} := \begin{pmatrix} 0 & 3 & 3 & 3 & 0 \\ 1 & 2 & 2 & 2 & 1 \\ 2 & 2 & 2 & 2 & 1 \\ 2 & 2 & 2 & 2 & 0 \\ 0 & 3 & 3 & 3 & 0 \end{pmatrix} \qquad \text{(M.140)}$$

```
ІМКпер := | for a ∈ 0.. N_A − 1                                        (М.141)
          |  for s ∈ 0.. S_CC − 1
          |    μ ← 0
          |    конф ← Конф_пер_{KP1_{s,a}−1, KP2_{s,a}−1}
          |    if конф ≠ ⟨4⟩
          |      for κ ∈ 0.. last(конф)
          |        μ_κ ← ІМК_{пошук[⟨МК_{конф_κ−1,a}⟩_s, ScF]_0}
          |    if конф = ⟨4⟩
          |      MAX ← max[⟨МК_{0,a}⟩_s, ⟨МК_{1,a}⟩_s, ⟨МК_{2,a}⟩_s]
          |      μ_0 ← ІМК_{пошук(MAX, ScF)_0}
          |    ІМКпер_{s,a} ← μ
          | ІМКпер
```

Зокрема, для першого аудіокадру (**a** = 0) і перших двох субсмуг (**c** = 0 і 1) МК кожної з трьох груп дорівнюють (див. також рис. 5.113):

$$\langle MK_{0,a}\rangle_0 = 0.00048828125000; \qquad \langle MK_{0,a}\rangle_1 = 0.00310039257963;$$

$$\langle MK_{1,a}\rangle_0 = 0.00390625000000; \qquad \langle MK_{1,a}\rangle_1 = 0.01240157071850;$$

$$\langle MK_{2,a}\rangle_0 = 0.12500000000000; \qquad \langle MK_{2,a}\rangle_1 = 0.07874566561843.$$

Що, згідно табл. А.1 Додатка, відповідає наступним ІМК:

$$пошук[\langle MK_{0,a}\rangle_0, ScF]_0 = 36; \qquad пошук[\langle MK_{0,a}\rangle_1, ScF]_0 = 28;$$

$$пошук[\langle MK_{1,a}\rangle_0, ScF]_0 = 27; \qquad пошук[\langle MK_{1,a}\rangle_1, ScF]_0 = 22;$$

$$пошук[\langle MK_{2,a}\rangle_0, ScF]_0 = 12; \qquad пошук[\langle MK_{2,a}\rangle_1, ScF]_0 = 14.$$

Різниці P_1 і P_2 при цьому дорівнюють відповідно +3 і –3, що у відповідності до табл. А.2 Додатка, визначає наступні класи:

$$KP1_{0,a} = 5, \quad KP2_{0,a} = 5; \qquad KP1_{1,a} = 5, \quad KP2_{1,a} = 5.$$

Така конфігурація передавання включає в себе індекси усіх трьох МК. Елемент масиву **ІМКпер** при цьому містить наступний вектор:

$$\text{ІМКпер}_{0,a} = \begin{pmatrix} 36 \\ 27 \\ 12 \end{pmatrix}; \qquad \text{ІМКпер}_{1,a} = \begin{pmatrix} 28 \\ 22 \\ 14 \end{pmatrix}.$$

Програмний модуль формування масиву ІВМК, що може заповнюватися з урахуванням конфігурації передавання, й у випадку обрання при динамічному розподілі бітів поточної субсмуги для кодування — (М.142). Візуалізований масив-результат передаваних кодів ІВМК, що має розмірність $S_{cc} \times N_A$, показаний на рис. 5.137.

$$\text{ІВМК} := \left| \begin{array}{l} \textbf{for}\ \ a \in 0..\ N_A - 1 \\ \quad \textbf{for}\ \ s \in 0..\ S_{CC} - 1 \\ \quad\quad \text{ІВМК}_{s,a} \leftarrow \text{ТІВМК}_{KP1_{s,a}-1,\ KP2_{s,a}-1} \\ \text{ІВМК} \end{array} \right. \qquad \text{(М.142)}$$

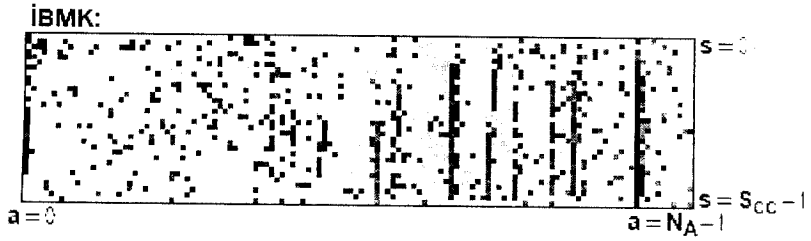


Рис. 5.137. Візуалізація масиву кодів ІВМК (білі ділянки відповідають коду «11», світло-сірі — «10», темно-сірі — «01», чорні — «00»)

Крок 6

Процедура динамічного розподілу бітів (див. розділи А.6, А.9 Додатка), що здійснюється в однойменному блоці ДРБ на рис. А.3 Додатка, починається з узгодження з фіксованою швидкістю передавання R і визначення кількості вільних бітів, доступних для кодування аудіоданих, — див. вирази (А.1) і (А.3) Додатка.

Зокрема, при швидкості $R = 64$ кбіт/с і тривалості аудіокадру $t_K = 24$ мс (у випадку $f_д = 48$ кГц) загальна кількість бітів, виділених на окремий кадр:

$$Б_\Sigma := R \cdot t_K = 64 \cdot 24 = 1536 \text{ біт/кадр}.$$

Загальна кількість бітів в окремому кадрі, необхідна для здійснення ДРБ субсмугових відліків, при $f_д = 48$ кГц визначається за табл. А.5 або А.6 Додатка, й у разі $R = 64 \geq 56$ кбіт/с становить:

$$Б_{розп.} := \text{if}(R \geq 56 \cdot 10^3,\ 88,\ 26) = 88 \text{ біт/кадр.}$$

Крім того, необхідна наступна кількість бітів для передавання службових або інших супутніх даних (див. розділи А.8, А.9 Додатка і, зокрема, рис. А.4):

- заголовок кожного аудіокадру: $Б_{заг.} := 32$ біт/кадр;
- контроль за наявністю помилок в аудіокадрі: $Б_{CRC} := 16$ біт/кадр;
- розширені дані *PAD* (поле може мати довільну довжину або ж узагалі не виділятися) — приймемо $Б_{XPAD} := 0$ біт/кадр;
- контроль за наявністю помилок у полі ІМК, які відносяться до субсмуг 2-х або 4-х різних спектральних груп (при цьому відповідно виділяються 16 або 32 біти) — приймемо $Б_{CRC\_iMK} := 32$ біт/кадр;
- фіксовані дані *PAD*: $Б_{FPAD} := 16$ біт/кадр.

Отже, початкова (гранична) кількість вільних для кодування бітів у кожному кадрі, згідно формул (А.1) і (А.3) Додатка, становить:

$$Б_{вільн.max} := Б_{\Sigma} - Б_{розп.} -$$
$$- (Б_{заг.} + Б_{CRC} + Б_{XPAD} + Б_{CRC\_iMK} + Б_{FPAD}) = 1352 \text{ біт/кадр.}$$

Загальна кількість субсмуг, на кодування відліків яких виділяються біти, визначається частотою $f_д$ і швидкістю R (див. табл. А.5, А.6 Додатка). При $f_д = 48$ кГц і $R = 64 \geq 56$ кбіт/с кодуються відліки субсмуг 0...26: $S'_{cc} = 27$.

Попередньо, перед початком процедури ДРБ, необхідні кількості бітів для кодування субсмугових відліків (як добутку кількості відліків або гранул субсмуги G_s на кількість бітів кодування відліку $Б_{відл.s}$), ІМК ($Б_{iMK_s}$) та ІВМК ($Б_{iBMK_s}$) кожної з $s := 0 .. S'_{cc}-1$ субсмуг встановлюються у нуль:

$$G_{o_s} = 0; \quad Б_{відл.o_s} = 0; \quad Б_{iMKo_s} = 0; \quad Б_{iBMKo_s} = 0.$$

При цьому початкове значення вільних бітів:

$$\max(Б_{відл.}) = Б_{вільн.max} -$$
$$- \sum_{s=0}^{S'_{cc}-1} \left(Á_{iMKo_s} + Á_{iBMKo_s} + G_{o_s} \cdot Á_{â^3äë.o_s} \right) = 1352 \text{ біт/кадр.}$$

Ітераційну процедуру ДРБ реалізує програмний модуль (М.143), що повертає п'ять масивів, розмірністю $S'_{cc} \times N_A$ кожний: 1) масив кількості виділених бітів для кодів ІВМК $Б_{iBMK}$; 2) масив кількості виділених бітів для ІМК $Б_{iMK}$; 3) масив кількості виділених бітів для кодових комбінацій відліків або гранул $Б_{відл.}$; 4) масив кількості від-

ліків або гранул, що представляються окремою кодовою комбінацією G; 5) масив індексів квантування, що визначають кількість кроків квантування окремих відліків всередині кодованої субсмуги, Q. Результати розподілу бітів для першого аудіокадру (a = 0), а також візуалізація підсумкових вищезазначених масивів наведені на рис. 5.138-5.142.

$$\begin{pmatrix} Б_{ІВМК} \\ Б_{ІМК} \\ Б_{відл.} \\ G \\ Q \end{pmatrix} := \text{for } a \in 0..N_A - 1 \quad (М.143)$$

$$Б_{ІВМК} \leftarrow Б_{ІВМКо},\ Б_{ІМК} \leftarrow Б_{ІМКо},\ Б_{відл.} \leftarrow Б_{відл.о},\ G \leftarrow G_o$$

$$\text{for } s \in 0..S'_{CC} - 1$$

$$MNR_s \leftarrow SNR_0 - SMR_{s,a},\ Q_s \leftarrow 0$$

$$Б_{вільн.} \leftarrow Б_{вільн.max} - \sum_{s=0}^{S'_{CC}-1} \left(Б_{ІВМК_s} + Б_{ІМК_s} + G_s \cdot Б_{відл._s}\right)$$

$$\text{while } mean(MNR) \neq 500$$

$$MNR_{min} \leftarrow \min(MNR),\ № \leftarrow пошук(MNR_{min}, MNR)_0$$

$$q \leftarrow Q_{№} + 1,\ Q_{№} \leftarrow q,\ S \leftarrow Nstep_{№,q}$$

$$MNR_{№} \leftarrow 500 \text{ if } S = 0$$

$$\text{if } S > 0$$

$$MNR_{№} \leftarrow SNR_{пошук(S,Step)_0} - SMR_{№,a}$$

$$Б_{відл.№} \leftarrow 5,\ G_{№} \leftarrow 12 \text{ if } S = 3$$

$$Б_{відл.№} \leftarrow 7,\ G_{№} \leftarrow 12 \text{ if } S = 5$$

$$Б_{відл.№} \leftarrow 10,\ G_{№} \leftarrow 12 \text{ if } S = 9$$

$$Б_{відл.№} \leftarrow \log_2(S+1),\ G_{№} \leftarrow 36 \text{ if } S \neq 3 \wedge S \neq 5 \wedge S \neq 9$$

$$MNR_{№} \leftarrow 500 \text{ if } Nstep_{№,q+1} = 0 \text{ if } q < cols(Nstep) - 1$$

$$MNR_{№} \leftarrow 500 \text{ if } q = cols(Nstep) - 1$$

$$Б_{ІМК_{№}} \leftarrow 6 \cdot rows(IMKnep_{№,a}),\ Б_{ІВМК_{№}} \leftarrow 2 \text{ if } q = 1$$

$$Б_{вільн.} \leftarrow Б_{вільн.max} - \sum_{s=0}^{S'_{CC}-1} \left(Б_{ІВМК_s} + Б_{ІМК_s} + G_s \cdot Б_{відл._s}\right)$$

$$\text{if } Б_{вільн.} \geq 0$$

$$Б^*_{ІВМК}{}^{(a)} \leftarrow Б_{ІВМК},\ Б^*_{ІМК}{}^{(a)} \leftarrow Б_{ІМК}$$

$$Б^*_{відл.}{}^{(a)} \leftarrow Б_{відл.},\ G^{*(a)} \leftarrow G,\ Q^{*(a)} \leftarrow Q$$

$$\text{break if } Б_{вільн.} \leq 0$$

$$\left(Б^*_{ІВМК}\ Б^*_{ІМК}\ Б^*_{відл.}\ G^*\ Q^*\right)^T$$

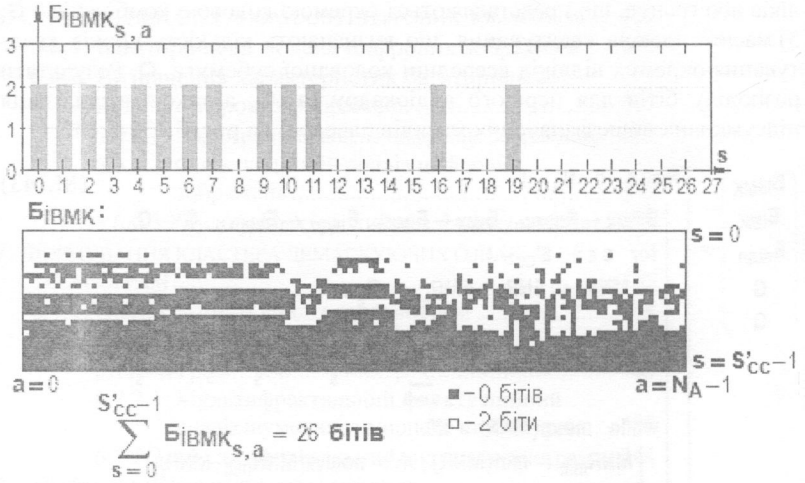


Рис. 5.138. Приклад результату виділення бітів для кодів ІВМК субсмуг аудіокадру, а також візуалізація масиву **Бівмк**

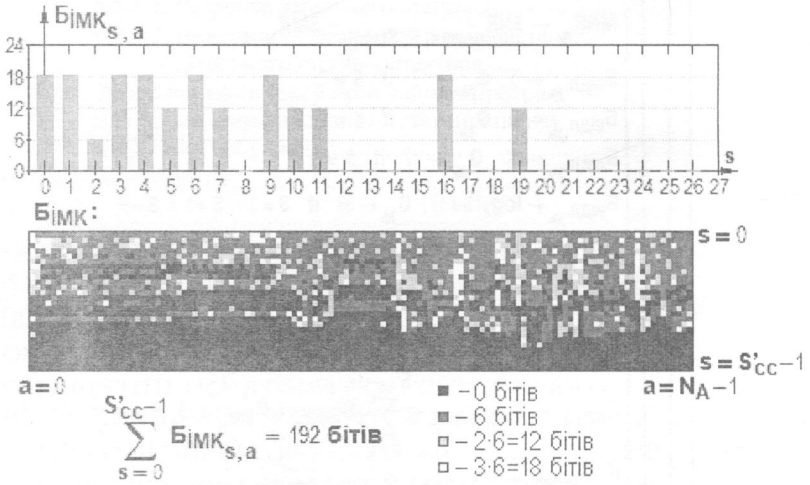


Рис. 5.139. Приклад результату виділення бітів для ІМК субсмуг аудіокадру, а також візуалізація масиву **Бімк**

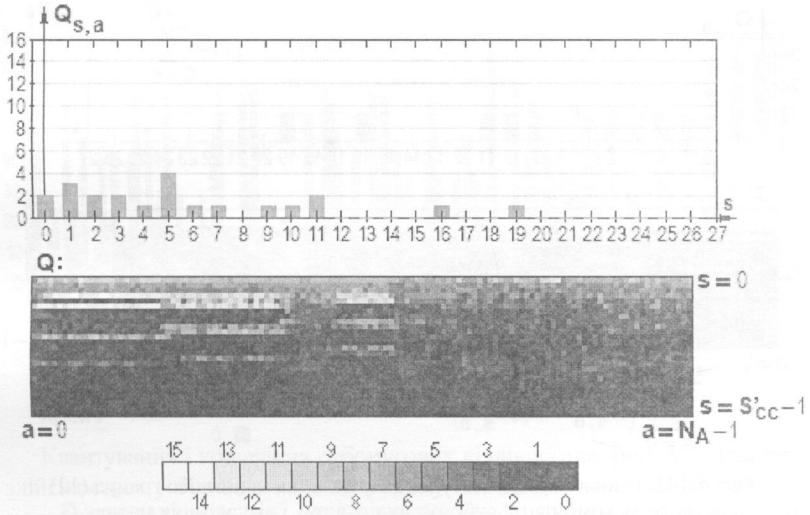


Рис. 5.140. Приклад одержаних індексів квантування і візуалізація масиву **Q**

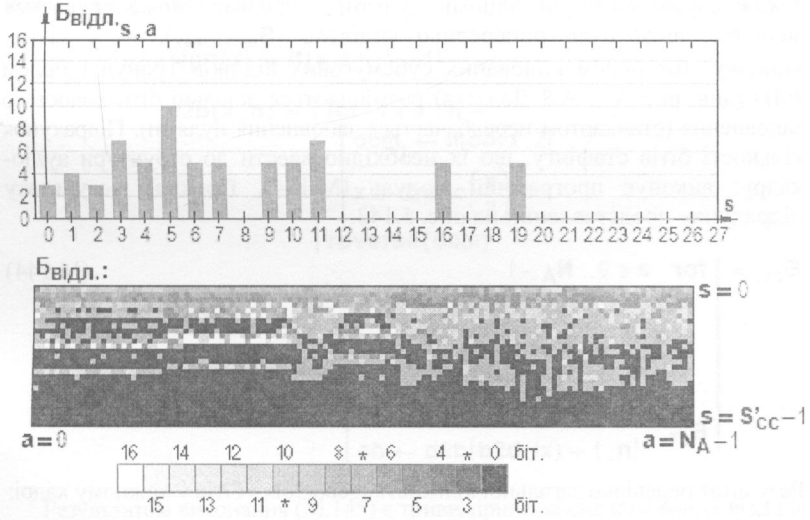


Рис. 5.141. Приклад результату виділення бітів для кодових комбінацій 36 відліків або 12 гранул субсмуг аудіокадру, і візуалізація масиву $Б_{відл.}$ (знаком «*» позначено випадки, коли для відліків і гранул властиві однакові довжини кодових слів)

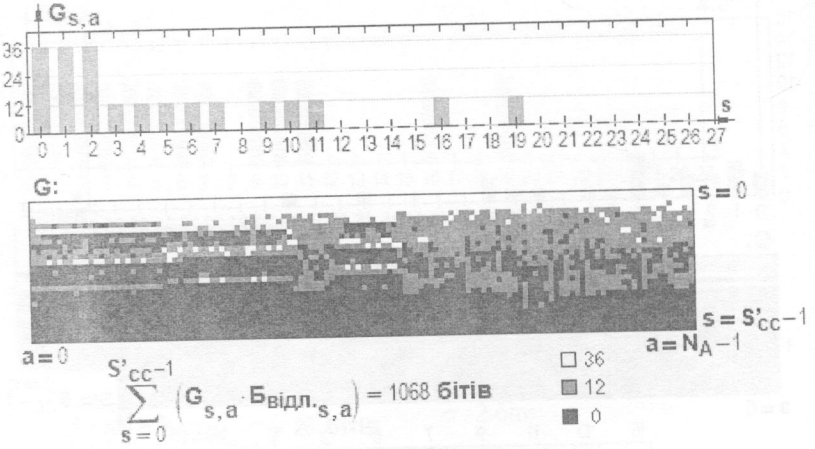


Рис. 5.142. Приклад результату виділення відліків (36) або гранул (12) для кодових комбінацій субсмуг аудіокадру, і візуалізація масиву **G**

Загальна кількість бітів, що фактично займаються аудіоданими, а також службовими чи іншими супутніми даними, може виявитися меншою, аніж було попередньо виділено ($Б_{вільн.max}$), — у цьому випадку між полем кодованих субсмугових відліків/гранул і полем *PAD* (див. рис. А.7, А.8 Додатка) розміщуються довільні біти холостого заповнення (стандартом передбачається заповнення нулями). Підрахунок кількості бітів стафінгу, що їх необхідно ввести до структури аудіокадру, виконує програмний модуль (М.144). Приклад результату підрахунку представлений на рис. 5.143.

$$Б_{ст.} := \left| \begin{array}{l} \text{for } a \in 0..N_A-1 \\ \quad Б_{ст.a} \leftarrow Б_{а.д.} - Б_{розп.} - \ldots \\ \qquad \ldots - \sum_{s=0}^{S'_{CC}-1}\left(Б_{ІВМК s,a} + Б_{ІМК s,a} + G_{s,a}\cdot Б_{відл.s,a}\right) \\ Б_{ст.} \end{array} \right. \qquad \text{(М.144)}$$

Результат перевірки загальної кількості «освоєних» бітів у кожному кадрі:

$$Б^*_{\Sigma} := \frac{\sum_{a=0}^{N_A-1}\left[Б_{розп.} + \sum_{s=0}^{S'_{CC}-1}\left(Б_{ІВМК s,a} + Б_{ІМК s,a} + G_{s,a}\cdot Б_{відл.s,a}\right) + Б_{ст.a} + Б_{заг.} + Б_{CRC} + Б_{XPAD} + Б_{CRC\_ІМК} + Б_{FPAD}\right]}{N_A} = 1536 \text{ бітів.}$$

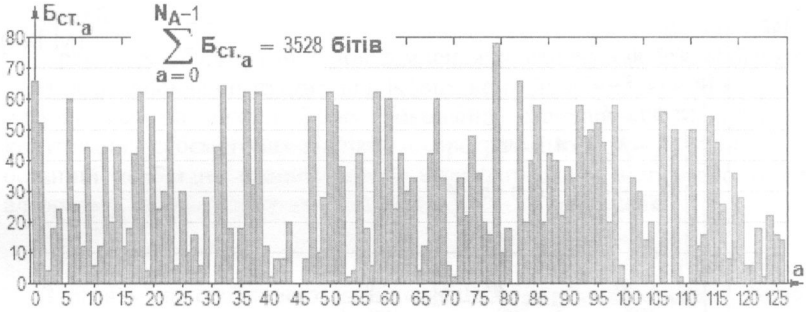


Рис. 5.143. Приклад покадрового підрахунку кількості бітів стафінгу

Крок 7

Квантування і кодування субсмугових відліків (див. рис. А.3 Додатка) здійснюється у відповідності до розділу А.7 за допомогою програмного модуля (М.145), в якому використано наступні підмодулі визначення знаку числа, зміну формату числа з десяткового на двійковий, з двійкового на десятковий, а також формування комплементарного (додаткового) коду двійкового числа:

$$\mathbf{sign}(x) := \mathbf{if}(x \geq 0, 1, -1)$$

$$\mathbf{d2b}(x, n) := \left| \begin{array}{l} \mathbf{for}\ \ i \in 0..n-1 \\ \quad \left| \begin{array}{l} ocm_i \leftarrow \mathbf{mod}(x, 2) \\ x \leftarrow \mathbf{trunc}\left(\frac{x}{2}\right) \end{array} \right. \\ \mathbf{reverse}(ocm) \end{array} \right.$$

$$\mathbf{b2d}(x) := \sum_{i=0}^{\mathbf{last}(x)} \left(x_i \cdot 2^{\mathbf{last}(x)-i} \right)$$

$$\mathbf{b2cb}(bx, n) := \left| \begin{array}{l} ix \leftarrow \overrightarrow{(\neg bx)} \\ cb \leftarrow \mathbf{d2b}(\mathbf{b2d}(ix) + 1, n) \end{array} \right.$$

Результатом виконання (М.145) є тривимірний масив кодованих відліків **Vq2** і допоміжний масив міток **F**, що мають розмірність $\mathbf{S'_{cc}} \times \mathbf{N_A}$. Кожен елемент масиву **Vq2**, індекси якого відповідають тим аудіокадровим субсмугам, на кодування яких за результатами ДРБ було виділено **n** бітів, являє собою вектор-стовпець **n**-символьної двійкової КК відповідного субсмугового відліку.

$$\begin{pmatrix} Vq2 \\ F \end{pmatrix} :=$$

$F \leftarrow 0 \cdot Q$

for $a \in 0 .. N_A - 1$

 for $s \in 0 .. S'_{CC} - 1$

  for $b \in 0 .. 35$

   $g1_b \leftarrow \left| V_{b,a} \right|_s$ if $b < 12$

   $g2_{b-12} \leftarrow \left| V_{b,a} \right|_s$ if $11 < b < 24$

   $g3_{b-24} \leftarrow \left| V_{b,a} \right|_s$ if $23 < b$

  $G1 \leftarrow \dfrac{g1}{\left| MK_{0,a} \right|_s}$, $G2 \leftarrow \dfrac{g2}{\left| MK_{1,a} \right|_s}$, $G3 \leftarrow \dfrac{g3}{\left| MK_{2,a} \right|_s}$

  $X\Sigma \leftarrow stack(G1, G2, G3)$, $q \leftarrow Q_{s,a}$, $S \leftarrow Nstep_{s,q}$, $n \leftarrow B_{відп_{s,a}}$

  if $S > 0$

   $step\_ind \leftarrow пошук(S, Step)_0 - 1$, $\Xi \leftarrow A_{step\_ind} \cdot X\Sigma + B_{step\_ind}$

   if $S \neq 3 \wedge S \neq 5 \wedge S \neq 9$

    for $b \in 0 .. 35$

     $зсув \leftarrow 2^{n-1} \cdot 0.999999999999$, $dec \leftarrow floor(зсув \cdot \Xi_b)$, $bin \leftarrow d2b(|dec|, n)$

     $bin \leftarrow b2cb(bin, n)$ if $sign(dec) = -1$

     $bin_0 \leftarrow bin_0 \oplus 1$, $Vq2_b \leftarrow bin$

    $F_{s,a} \leftarrow$ 'окремо'

   otherwise

    $b \leftarrow 0$, $зсув \leftarrow$ if$(S = 3, 1.999999999999$, if$(S = 5, 3.999999999999, 7.999999999999))$

    $\nu \leftarrow$ if$(S = 3, 2,$ if$(S = 5, 3, 4))$

    while $b < 34$

     $dec\_x \leftarrow floor(зсув \cdot \Xi_b)$, $dec\_y \leftarrow floor(зсув \cdot \Xi_{b+1})$, $dec\_z \leftarrow floor(зсув \cdot \Xi_{b+2})$

     $bin\_x \leftarrow d2b(|dec\_x|, \nu)$, $bin\_y \leftarrow d2b(|dec\_y|, \nu)$, $bin\_z \leftarrow d2b(|dec\_z|, \nu)$

     $bin\_x \leftarrow b2cb(bin\_x, n)$ if $sign(dec\_x) = -1$

     $bin\_y \leftarrow b2cb(bin\_y, n)$ if $sign(dec\_y) = -1$

     $bin\_z \leftarrow b2cb(bin\_z, n)$ if $sign(dec\_z) = -1$

     $bin\_x_0 \leftarrow bin\_x_0 \oplus 1$, $bin\_y_0 \leftarrow bin\_y_0 \oplus 1$, $bin\_z_0 \leftarrow bin\_z_0 \oplus 1$

     $x \leftarrow b2d(bin\_x)$, $y \leftarrow b2d(bin\_y)$, $z \leftarrow b2d(bin\_z)$

     $v \leftarrow$ if$(S = 3, 9 \cdot z + 3 \cdot y + x,$ if$(S = 5, 25 \cdot z + 5 \cdot y + x, 81 \cdot z + 9 \cdot y + x))$

     $bin\_v \leftarrow d2b(v, n)$, $Vq2_b \leftarrow bin\_v$, $Vq2_{b+1} \leftarrow bin\_v$, $Vq2_{b+2} \leftarrow bin\_v$

     $b \leftarrow b + 3$

    $F_{s,a} \leftarrow$ 'гранула'

  $Vq2\Sigma_{s,a} \leftarrow Vq2$

$\begin{pmatrix} Vq2\Sigma \\ F \end{pmatrix}$

(M.145)

Крок 8

Процедура вбудовування повідомлення до стиснутого аудіосигналу полягає у заповненні пустих відлікових контейнерів, сукупність яких одержується на виході блоку виконання стандартного кодування квантованих субсмугових відліків — програмного модуля (М.145), — останній необхідно певним чином модифікувати у відповідності до алгоритму, що реалізуватиме заданий метод стеганографії.

Алгоритм заміни найменших значущих бітів

З огляду на особливості заміни НЗБ, передбачається, що даний алгоритм доцільно застосовувати при реалізації методів приховання повідомлень додаткової інформації значного об'єму (потребують високої ППЗ) і відкритого характеру (тобто не вимагають стійкого захисту від стороннього доступу).

Адаптований спосіб заміни НЗБ у кодових комбінаціях, що відповідають субсмуговим відлікам або гранулам програмних потоків кадрів *MPEG Layer II*, реалізується на основі програмного модуля (М.145), шляхом внесення до його точок, позначених як **«A1»** і **«A2»**, відповідних процедурі (5.59) доповнень, представлених сегментами (М.146). При цьому біти повідомлення **Mbin** (вектора довжиною L_M) послідовно заміщують собою НЗБ кодів лише тих відліків і гранул кожного кадру, на кодування яких за результатом ДРБ було виділено достатню для створення комфортного ефекту маскування кількість бітів (більше заздалегідь обраного порогового значення λ).

$$НЗБ\left(bin \vee bin\_v, \mu\right) = Mbin_{\mu}, \quad 0 \le \mu < L_M, \quad index(НЗБ) > \lambda. \qquad (5.59)$$

«A1»:
```
⋮
if  μ < L_M  if  n > λ
    bin_{n-1} ← Mbin_μ
    μ ← μ + 1
⋮
```
(М.146, *а*)

«A2»:
```
⋮
if  μ < L_M  if  n > λ
    bin_v_{n-1} ← Mbin_μ
    μ ← μ + 1
⋮
```
(М.146, *б*)

При виконанні (М.145), доповненого у відповідності до (М.146), змінюватиметься лише вміст векторів, розрядність яких задовольняє умові $n > \lambda$. Розмірності вихідних масивів **Vq2** (кодовані відліки) та **F**

(мітки) змін при цьому не зазнають. Вектор **Mbin** є результатом бінаризації і, за потреби, розгортання у вектор повідомлення **M**, що надходить у систему попередньої обробки з виходу визначеного джерела (див. рис. 2.3) і може мати довільний первинний формат представлення мультимедійних даних (текстовий документ, звук, відео, програмний файл тощо)[XXIII].

Алгоритм відносної заміни квантованих субсмугових відліків

З розділу 3 відомо, що для забезпечення прийнятної стійкості стеганограми до можливих трансформацій мультимедійного контейнера, остання має бути вбудована до тих його елементів, що не піддаються впливові визначеного набору спотворень заданого ступеня. У випадку використання в якості контейнерів стиснутих потоків аудіосигналів найімовірнішим типом загрози вмісту стеганограми, прихованої у субсмугових відліках, є повторне (нове) стиснення з проміжними етапами реквантування, регенерації і субсмугового синтезу при формуванні вихідного (відновленого) ІКМ-сигналу, а також нового субсмугового аналізу, розрахунку МК, обчислення ПАМ, виконання ДРБ, нормування, квантування і кодування субсмугових відліків при формуванні потоку з іншим (або навіть таким самим) рівнем стиснення. В останньому випадку етапи нового стиснення, навіть за незмінної швидкості передавання аудіоданих ***R***, можуть мати своїм результатом дещо відмінні бітові послідовності, що спричиняється, зокрема, достатньо великою кількістю існуючих різновидів ПАМ.

Стеганоалгоритм може бути достатньо стійким до наступного стиснення мультимедійних бітових послідовностей (у тому числі й аудіо), якщо враховуватиме особливості перспективного стиснення. При цьому алгоритм, до основи якого закладено використання контейнерів (заповнених субсмугових відліків), отриманих, наприклад, на основі стандартної ПАМ № 2 [126,127], не обов'язково буде стійким до наступного стиснення на основі ПАМ № 1 навіть того ж стандарту[XXIV].

[XXIII] Наприклад, програмний модуль бінаризації повідомлення, що первинно має текстовий формат (файл "M.txt"):

```
Mbin := | M ← READBIN("M.txt", "byte", 1)
        | Mvec ← str2vec(M)
        | Mbin ← d2b(Mvec_0, 8)
        | for m ∈ 1.. last(Mvec)
        |   Mbin ← stack(Mbin, d2b(Mvec_m, 8))
        | Mbin
```

[XXIV] Взагалі, питання щодо існування стійких стеганографічних перетворень, які були б інваріантними до можливих у подальшому алгоритмів стиснення із втратами, залишається відкритим [5].

З урахуванням представленої у п. 5.3.3 моделі оцінки стійкості стеганографічної системи за наявності у каналі зв'язку атаки повторним стисненням, повідомлення **M** передається стеганоканалом, що має два джерела «шуму»: субсмугові відліки зі складу окремого аудіосигналу, а також атака повторним стисненням **Θ** — див. рис. 5.27. При цьому $\mathbf{S}^{\mathbf{V}}$ і $\mathbf{M}^{*}$ — відповідно спотворені стеганоконтейнер і повідомлення (їхня оцінка).

Реальні аудіосигнали являють собою випадкові процеси з нерівномірно розподіленими значеннями величин [131, 132, 140]. Отриману методом так званого балансного регулювання криву порогу чутності тональних звуків, виміряну в умовах вільного поля, приведено на рис. 5.144 [141, 142]. Як видно, поріг чутності в залежності від частоти змінюється у широких межах, а найбільшу чутливість ССЛ має в області частот 2500...3500 Гц, де поріг чутності має мінімальну величину. Крім того, музичні і мовні сигнали займають лише частину області чутності (як по частоті, так і за амплітудою). Субсмуги в частотній області від 50 до 5000 Гц містять основну енергію аудіосигналу (за рівнем звукового тиску — від 30 до 100 дБ) і, таким чином, щодо приховуваного повідомлення носять шумовий (маскуючий) характер. Високо- та інфрачастотні субсмуги спектру звукового сигналу найбільшим чином піддаються впливу з боку різноманітних алгоритмів обробки, таких як, наприклад, компресія. У результаті застосування алгоритмів аудіостиснення зазначені області чутності ще більше обмежуються.

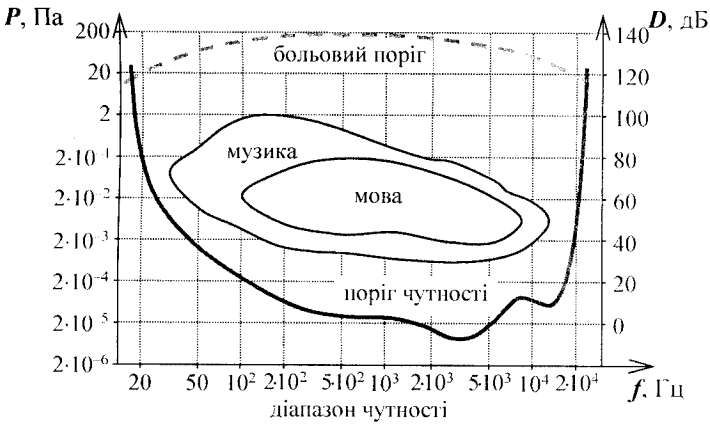


Рис. 5.144. Криві порогу чутності і больового порогу, а також області чутності музики і мови

Отже, у разі необхідності одержання стійкої до повторного стиснення стеганосистеми є очевидним, що вбудовування необхідно здійснювати до середньочастотних субсмуг спектру сигналу (за виключенням, можливо, ділянки 2...5 кГц). Стеганоалгоритм, до основи якого закладено даний принцип, доцільно застосовувати при реалізації методів приховання повідомлень на зразок ЦВЗ, що мають невеликий об'єм (а, отже, не потребують високої ППЗ), вимагають організації умовного допуску (потребують захисту від стороннього доступу) і мають бути в достатній мірі стійкими до визначених типів атак (руйнування повідомлення припускається лише за достатньої деградації контейнера) [3].

Пропонований нижче спосіб вбудовування бітів ЦВЗ ґрунтується також на тих фактах, що ССЛ розрізнює зміни у фазі сигналу слабше, ніж зміни його амплітуди або частоти [140]. Крім того, — фазові компоненти аудіосигналу, в значній мірі зумовлюючи загальну якість його звучання, є більш стійкими до впливу шуму атаки Θ (наприклад, у результаті повторного стиснення). Таким чином, якщо вносити ЦВЗ у фазові компоненти, успіх у видаленні ЦВЗ неминуче призведе до значного руйнування самого контейнера [3,5,132].

За основу алгоритму вбудовування ЦВЗ до стиснутого аудіосигналу (а в більш загальному випадку — повідомлень, пов'язаних з цифровим аудіопотоком) покладено ідею реалізації методів [86,87] з відповідною адаптацією останніх до особливостей субсмугових відліків потоку *MPEG Layer II*, використовуваних в якості контейнерів. При цьому алгоритм передбачає маніпуляцію фаз відліків-коефіцієнтів **V** субсмугового аналізу сигналу у БФА шляхом відносної заміни їх нормованих ($\mathbf{X\Sigma}$) і квантованих (Ξ) версій у парі кодованих за результатом ДРБ в кадрі a середньочастотних субсмуг $s1$ і $s2$, $s1 \neq s2$ (обрання яких може, наприклад, ініціюватися секретними ключами **K1** і **K2**), реалізуючи симетричну модель стеганосистеми змішаного типу (див. п. 2.4.4).

Вбудовування бітів ЦВЗ **W** здійснюється згідно процедури

$$\forall b \begin{cases} \Xi_b(s1 = \mathbf{K1}_a) > \Xi_b(s2 = \mathbf{K2}_a) & \text{при } w = \mathbf{Wbin}_a = 0; \\ \Xi_b(s1 = \mathbf{K1}_a) < \Xi_b(s2 = \mathbf{K2}_a) & \text{при } w = \mathbf{Wbin}_a = 1, \end{cases} \tag{5.60}$$

виконання якої покладено на програмний модуль (М.145) із внесеною до позиції **«B»** модифікацією, сутність якої пояснює сегмент (М.147). Розмірності вихідних масивів кодованих відліків **Vq2** і міток **F** при цьому не зазнають змін. Змінюється лише вміст векторів у складі масиву **Vq2**. Вектор **Wbin** — це бінаризоване представлення повідомлення **M**, що надходить до системи попередньої обробки з виходу

деякого джерела ЦВЗ (див. рис. 2.3) і може мати довільний первинний формат даних (рядок тексту, зображення-логотип тощо)[XXV].

«B»
$$\begin{aligned}
&\text{if } s = K1_a \\
&\quad \text{for } b \in 0..35 \\
&\quad\quad \Xi_b \leftarrow A_{step_ind} \cdot \left|X\Sigma_b\right| + B_{step_ind} \quad \text{if } Wbin_a = 0 \\
&\quad\quad \Xi_b \leftarrow -A_{step_ind} \cdot \left|X\Sigma_b\right| + B_{step_ind} \quad \text{if } Wbin_a = 1 \\
&\text{if } s = K2_a \\
&\quad \text{for } b \in 0..35 \\
&\quad\quad \Xi_b \leftarrow -A_{step_ind} \cdot \left|X\Sigma_b\right| + B_{step_ind} \quad \text{if } Wbin_a = 0 \\
&\quad\quad \Xi_b \leftarrow A_{step_ind} \cdot \left|X\Sigma_b\right| + B_{step_ind} \quad \text{if } Wbin_a = 1
\end{aligned} \tag{M.147}$$

Алгоритм заміни найменших значущих бітів у псевдовипадково обраних субсмугових відліках

Для передавання повідомлень додаткової інформації, що потребує обов'язкового захисту стеганошляху (сервісна інформація за передплатою, конфіденційна інформація тощо) і, як правило, наявності високої ППЗ стеганографічного каналу, може бути застосований класичний високошвидкісний алгоритм заміни НЗБ (5.59) у реалізації (M.146), але з забезпеченням додаткової можливості обрання в якості контейнера псевдовипадкового відліку $\mathbf{C}_\beta$ з усієї множини наявних в окремому аудіокадрі a [XXVI].

Сутність пропонованого для використання алгоритму заміни НЗБ кодових комбінацій псевдовипадково обраних субсмугових відліків полягає у тому, що генератор ПВП створює послідовність індексів

[XXV] Програмний модуль бінаризації ЦВЗ, що первинно має графічний формат (файл "W.bmp"), наприклад, може виглядати так:

Wbin :=
$$\begin{aligned}
&W \leftarrow \text{READBMP}(\text{"W.bmp"}) \\
&Wbin \leftarrow W^{(0)} \\
&\text{for } i \in 1..\text{cols}(W) - 1 \\
&\quad Wbin \leftarrow \text{stack}(Wbin, W^{(i)}) \\
&\frac{Wbin}{255}
\end{aligned}$$

[XXVI] Модифікований подібним чином алгоритм заміни НЗБ стає подібним до алгоритму псевдовипадкової перестановки, представленому у [79] і розглянутому на прикладі графічного контейнера у підп. 5.3.2.3.

$\beta_0, \beta_1, \ldots, \beta_{11(35)}$ і зберігає μ-й біт повідомлення ***Mbin***, що має довжину L_M, у відліку з індексом $\beta_{mod[\mu,12(36)]}$. А, власне, вбудовування бітів повідомлення здійснюється відповідно до процедури (5.59).

Нехай $N_{НЗБ}(s,a)$ — загальна кількість доступних НЗБ у субсмузі s кадру a (див. крок № 6),

$$N_{НЗБ}(s,a) = G_{s,a}, \text{ причому } G_{s,a} = (12; 36). \qquad (5.61)$$

Тоді, якщо $P[N_{НЗБ}(s,a)]$ — це перестановка чисел $\{1, 2, \ldots, N_{НЗБ}(s,a)\}$, то L_M бітів конфіденційного повідомлення $Mbin_\mu$ (де $\mu = 0 \ldots L_M - 1$) можна просто вбудовувати замість НЗБ контейнерів з індексами $P[N_{НЗБ}(s,a)]_0$, $P[N_{НЗБ}(s,a)]_1$, $P[N_{НЗБ}(s,a)]_2$, ..., $P[N_{НЗБ}(s,a)]_{mod[\mu,12(36)]}$. При цьому сама функція P має бути псевдовипадковою, забезпечуючи обрання контейнерів-відліків у межах окремо взятого кадру a приблизно рівноймовірним чином. Крім того, вона також повинна залежати від секретного ключа $k \in \mathbf{K}$, виробляючи для кожного нового k унікальні псевдовипадкові перестановки чисел $\{1, 2, \ldots, N_{НЗБ}(s,a)\}$. Для відсутності перетинань (див. пояснення до виразів (2.1) і (2.2)) доцільно застосовувати генератори ПВП без повторюваності — наприклад, на основі на генератора псевдовипадкової функції [98].

Виконання процедури (5.61) із псевдовипадковим обранням для заповнення контейнерів-відліків $\mathbf{C}_\beta$ бітами послідовно зчитуваного повідомлення (або, що те саме, — псевдовипадковим обранням бітів вбудовуваного повідомлення $Mbin_\mu$, якщо індекси контейнерів перебираються послідовно) покладено на основний програмний модуль — (М.145) — із внесеною у позицію «С» модифікацією, сутність якої пояснює сегмент (М.148). При цьому внесення бітів повідомлення виконується лише до субсмуг **s** (починаючи з певної ζ-ї) кадру **a**, відліки яких кодуються окремо ($F_{s,a}$ = "окремо"), а розрядність їх кодування **n** перевищує поріг λ.

```
⋮
if  n > λ ∧ s ≥ ζ                                          (М.148)
    μ ← пошук(b , β)_0 + Δμ   if  b < rows(β)
    bin_{n-1} ← Mbin_μ   if  μ < L_M
    Vq2_b ← bin
Δμ ← Δμ + rows(β) − ζ   if   n > λ ∧ s ≥ ζ
F_{s,a} ← 'окремо'
⋮
```

Крок 9

Для одержання на виході стеганографічного аудіокодера бітового потоку кадрів (крок 9.4), на вхід формувача останніх для мультиплексування надходять (див. розділи А.8–А.10 Додатка):

- заголовок кадру (крок 9.1) і кодове слово *CRC* (крок 9.2);
- дані про розподіл бітів по субсмугам, ІВМК, ІМК (кроки 5, 6);
- потік пустих і заповнених субсмугових відліків-контейнерів (кроки 7, 8);
- поле фіксованих даних *F-PAD* і, за необхідності, контрольна сума *CRC* ІМК [XXVII] (крок 9.3).

Крок 9.1

Заголовок аудіокадру, згідно підрозділу А.9.1 Додатка, повинен містити:

- 12-бітову синхрогрупу: **sync** := (1 1 1 1 1 1 1 1 1 1 1 1);
- біт-ідентифікатор стандарту аудіокодування — для *ISO/IEC 11172-3* **ID** := (1);
- 2 біти індикатора потоку — для *Layer II* **layer** := (1 0);
- біт мітки наявності захисту контрольною сумою *CRC*: **protect** := (0);
- 4-бітовий ІІШП — для **R** = 64 кбіт/с **bitrate** := (0 1 0 0);
- 2 біти індексу частоти дискретизації — для $\mathbf{f_д}$ = 48 кГц **samplfr** := (0 1);
- біт заповнення: **padding** := (0);
- біт права особистого користування (не розпізнається): **private** := (0);
- 2 біти індексу аудіорежиму — для одноканального режиму **mode** := (1 1);
- 2 біти індексу розширення аудіорежиму: **ext** := (0 0);
- біт авторських прав на цифровий аудіопотік: © := (0);
- біт оригінальності або копії аудіопотоку: **orig/copy** := (0);
- 2 біти індексу передспотворень: **emphasis** := (0 0).

Об'єднання елементів заголовка до спільного масиву й отриманий при цьому результат показані на рис. 5.145.

$$\mathbf{Header} := \mathbf{augment}(\mathbf{sync}, \mathbf{ID}, \mathbf{Layer}, \mathbf{protect}, \mathbf{bitrate}, \mathbf{samplfr}, \mathbf{padding}, \mathbf{private}, \mathbf{mode}, \mathbf{ext}, ©, \mathbf{orig/copy}, \mathbf{emphasis})^{T}$$

$$\mathbf{Header}^{T} = (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0)$$

Рис. 5.145. Результат формування заголовка аудіокадру

[XXVII] Поле розширених даних *X-PAD*, через зниклу у цьому необхідність, до структури аудіокадру не додається ($\boldsymbol{Б}_{XPAD} = 0$ біт/кадр).

Крок 9.2

У відповідності до підрозділу А.9.2 Додатка, 3-й і 4-й байти в складі заголовка кожного окремого кадру, а також відповідні дані ДРБ та ІВМК захищаються 16-бітовим словом парності *CRC*. Програмний модуль, що формує захищувану послідовність, — (М.149). Приклад отриманої при цьому послідовності для першого (**a** = 0) аудіокадру наведений на рис. 5.146.

```
CRC_in := | for a ∈ 0.. N_A − 1                                        (M.149)
          |   input ← augment(bitrate. samplfr. padding. private.
          |                    mode. ext. ©. orig/copy. emphasis)^T
          |   for s ∈ 0.. S'_CC − 1
          |     q ← Q_s,a.   n ← Б_розп,s
          |     bit_alloc ← d2b(q. n)
          |     input ← stack(input. bit_alloc)
          |   for s ∈ 0.. S'_CC − 1
          |     if Б_IBMK_s,a = 2
          |       scf_inf ← d2b(IBMK_s,a. 2)
          |       input ← stack(input. scf_inf)
          |   CRC_in_a ← input
          | CRC_in
```

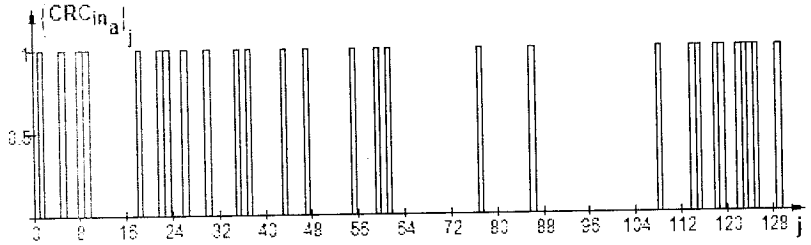


Рис. 5.146. Приклад індексної діаграми захищуваної кодом *CRC*-послідовності

Обчислення слова *CRC* здійснюється за допомогою 16-розрядного (n = 16) регістра зсуву, представленого на рис. В.1 Додатка, і визначається генеруючим поліномом $G_{16}(x)$ (В.1), (В.3). Первинному стану регістра відповідає запис до всіх його розрядів «1». Наведена в Додатку В процедура обчислення *CRC* реалізується програмним модулем (М.150).

$$
\begin{aligned}
&\mathbf{CRC_{out}(a)} := \Big|\ \mathbf{g} \leftarrow (\underset{\mathbf{g}_2}{1}\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ \underset{\mathbf{g}_{15}}{1})^{T} \\
&\quad \mathbf{B} \leftarrow (1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1)^{T} \\
&\quad \mathbf{input} \leftarrow \mathbf{CRC_{in_a}} \\
&\quad \mathbf{for}\ \ j \in 0..\ \mathbf{last(input)} \\
&\quad\quad \mathbf{bufer}_0 \leftarrow B_0,\ \ B_0 \leftarrow B_{n-1} \oplus \mathbf{input}_j \\
&\quad\quad \mathbf{for}\ \ i \in 1..\ n-1 \\
&\quad\quad\quad \mathbf{bufer}_i \leftarrow B_i,\ \ B_i \leftarrow B_0 \cdot \mathbf{g}_i \oplus \mathbf{bufer}_{i-1} \\
&\quad \mathbf{CRC_{out}} \leftarrow \mathbf{B}
\end{aligned}
\qquad \text{(M.150)}
$$

При цьому одиничні коефіцієнти $\mathbf{g}_i$ полінома $G_{16}(x)$ зведені до вектора $\mathbf{g}$, а первинний стан регістра визначається одиничним вектором $\mathbf{B}$ на початку програмного модуля. У підсумку, для кожного кадру одержується 16-елементний вектор контрольної суми. Зокрема, для аудіокадру $\mathbf{a} = 0$:

$$\mathbf{CRC_{out}}(0)^{T} = (0\,1\,1\,0\,1\,1\,0\,1\,0\,1\,0\,0\,1\,1\,0\,0).$$

Крок 9.3

Оскільки процедура формування полів *CRC* ІМК та *F-PAD* перебуває за межами даного дослідження, враховуватимемо лише їхню наявність і довжину в загальній структурі аудіокадрів (див. крок 6): $\mathbf{Б_{CRC\_ІМК}} = 32$ біт/кадр, $\mathbf{Б_{FPAD}} = 16$ біт/кадр. А відповідні поля заповнюємо нулями:

$$
\begin{aligned}
&\mathbf{CRCІМК} := \Big|\ \mathbf{for}\ \ i \in 0..\ \mathbf{Б_{CRC\_ІМК}} - 1 \\
&\quad\quad \mathbf{CRCІМК}_i \leftarrow 0 \\
&\quad \mathbf{CRCІМК}
\end{aligned}
\qquad \text{(M.151)}
$$

$$
\begin{aligned}
&\mathbf{FPAD} := \Big|\ \mathbf{for}\ \ i \in 0..\ \mathbf{Б_{FPAD}} - 1 \\
&\quad\quad \mathbf{FPAD}_i \leftarrow 0 \\
&\quad \mathbf{FPAD}
\end{aligned}
$$

Крок 9.4

Об'єднання бітових послідовностей заголовка **Header**, слова контрольної суми $\mathbf{CRC_{out}(a)}$, даних ДРБ **BitAlloc**, ІВМК **ScFinf** та ІМК **ScFind**, пустих і заповнених контейнерів (кодових комбінацій) субсмугових відліків **Vq2**, бітів стафінгу (якщо їхнє введення є необхідним), службових даних **CRCІМК** та **FPAD** здійснюється програмним модулем (М.152).

```
BitStream := | BS ← stack(|Header , CRC_out(0)|)
             | for a ∈ 0.. N_A − 1
             |   for s ∈ 0.. S'_CC − 1
             |   |  q ← Q_s,a ,  n ← Б_розп.,s
             |   |  BitAlloc ← d2b(q , n)
             |   |  BS ← stack(BS , BitAlloc)
             |   for s ∈ 0.. S'_CC − 1
             |     if Б_iBMK_s,a = 2
             |     |  ScFinf ← d2b(|iBMK_s,a , 2|)
             |     |  BS ← stack(BS , ScFinf)
             |   for s ∈ 0.. S'_CC − 1
             |     if Б_iMK_s,a = 6 ∨ Б_iMK_s,a = 12 ∨ Б_iMK_s,a = 18
             |     |  ScFind ← iMKnep_s,a
             |     |  for i ∈ 0.. last(ScFind)
             |     |    index ← d2b(|ScFind_i , 6|)
             |     |    BS ← stack(BS , index)
             |   b ← 0
             |   while b < 34
             |     |  for s ∈ 0.. S'_CC − 1
             |     |    if F_s,a = 'окремо'
             |     |      for j ∈ 0.. 2
             |     |        BS ← stack[BS , |Vq2_s,a|_j+b]
             |     |    if F_s,a = 'гранула'
             |     |      BS ← stack[BS , |Vq2_s,a|_b]
             |     |  b ← b + 3  if s = S'_CC − 1
             |   for r ∈ 0.. Б_ст.,a − 1   if Б_ст.,a > 0
             |     BS ← stack[BS , |0|]
             |   BS ← stack(|BS, CRCiMK, FPAD, Header, CRC_out(a)|)
             | BS
```
(M.152)

Результатом виконання (М.152) є вектор бітового потоку **BitStream** довжиною $N_A \cdot Б_\Sigma$ (остання для розгляданого прикладу становить: 127·1536 = 195072 біти).

Для форматування отриманого бітового потоку у відповідності до стандартної байтової структури з наступним її записом до кінцевого файлу аудіоданих, стиснутих за алгоритмом стандарту *MPEG-1 Layer II*, використовується програмний модуль (М.153). Розмір отриманого в прикладі файлу становить 24384 байти.

$$\text{Byte\_formater}(x) := \left| \begin{array}{l} \text{for } i \in 0..\dfrac{\text{rows}(x)}{8} - 1 \\ \quad \text{Byte}_i \leftarrow \text{b2d}(\text{subvec}(x, i \cdot 8, i \cdot 8 + 7)) \\ \text{Byte} \end{array} \right. \qquad \text{(M.153)}$$

$$\text{WRITEBIN}(\text{"S.mp1"}, \text{"byte"}, 1) := \text{Byte\_formater}(\text{BitStream})$$

Крок 10

Процедура видобування бітів прихованого повідомлення для застосованих на кроці 8 алгоритмів є оберненою процедурі вбудовування і ґрунтується на обранні й дослідженні заповнених контейнерів (субсмугових відліків).

Бітове форматування зчитаного файлу стиснутого аудіосигналу виконується програмним модулем (М.154).

$$\text{BitStream}^* := \left| \begin{array}{l} \text{ByteStream}^* \leftarrow \text{READBIN}(\text{"S.mp1"}, \text{"byte"}, 1) \\ \text{BitStream}^* \leftarrow \text{d2b}(\text{ByteStream}^*_0, 8) \\ \text{for } i \in 1..\text{last}(\text{ByteStream}^*) \\ \quad \text{BitStream}^* \leftarrow \text{stack}(\text{BitStream}^*, \text{d2b}(\text{ByteStream}^*_i, 8)) \\ \text{BitStream}^* \end{array} \right. \qquad \text{(M.154)}$$

Аналіз заголовка кожного аудіокадру, починаючи з найпершого ($a = 0$) повертає наступні вихідні дані, необхідні для подальшого розщеплення (при цьому вважається, що при перевірці кодом *CRC* помилок не виявлено):

- аудіорежим: одноканальний, $N^*_к := 1$;
- частота дискретизації: $f^*_д := 48$ кГц;
- сумарна бітова швидкість при заданій частоті: $R^* := 64$ кбіт/с.

Оскільки кількість відліків в окремому кадрі $S^*_к = 1152$, то тривалість аудіокадру становить $t^*_к := S^*_к / f^*_д = 24$ мс, загальна кількість бітів, що була виділена на кадр: $Б^*_Σ := R^* \cdot t^*_к = 1536$ біт/кадр. При цьому для передавання службових та інших супутніх даних (*PAD*) було виділено наступні кількості бітів, які або задані стандартом, або визначаються за заголовками полів даних:

- передавання заголовку: $Б^*_{заг.} := 32$ біт/кадр;

- контроль помилок у кадрі: $Б^*_{CRC} := 16$ біт/кадр;
- розширені дані *PAD*: $Б^*_{XPAD} := 0$ біт/кадр;
- контроль помилок у полі ІМК: $Б^*_{CRC\_iмк} := 32$ біт/кадр;
- фіксовані дані *PAD*: $Б^*_{FPAD} := 16$ біт/кадр.

Загальна кількість бітів в окремому кадрі, якими описується ДРБ для кодування субсмугових відліків, з урахуванням $f^*_д = 48$ кГц і $R^* = 64$ кбіт/с, визначається згідно табл. А.5 Додатка і становить:

$$Б^*_{розп.} := \mathbf{if}(R^* \geq 56 \cdot 10^3, 88, 26) = 88 \text{ біт/кадр.}$$

Порядковий номер верхньої субсмуги, відліки якої ще могли бути при цьому кодовані,

$$S^{*\prime}_{cc} := \mathbf{if}(R^* \geq 56 \cdot 10^3, 27, 8) = 27.$$

Загальна кількість кадрів в аналізованому бітовому потоці стиснутого аудіосигналу:

$$N^*_A := \mathbf{rows}(\mathbf{BitStream}^*) / Б^*_\Sigma = 195072 / 1536 = 127.$$

Розщеплення бітового потоку $\mathbf{BitStream}^*$ на окремі кадри **ASK** виконує програмний модуль (М.155).

$$\mathbf{ASK} := \left| \begin{array}{l} \mathbf{for}\ \ a \in 0 .. N^*_A - 1 \\ \quad ASK_a \leftarrow \mathbf{subvec}\left[\mathbf{BitStream}^*,\ a \cdot Б^*_\Sigma,\ (a+1) \cdot Б^*_\Sigma - 1\right] \\ ASK \end{array} \right. \qquad \text{(М.155)}$$

Програмний модуль (М.156) повертає масив $\mathbf{Q}^*$ індексів квантування, що визначають кількість кроків, з якими були квантовані окремі відліки всередині кодованої субсмуги. Розмірність масиву: $S^{*\prime}_{cc} \times N^*_A$.

$$\mathbf{Q}^* := \left| \begin{array}{l} \mathbf{for}\ \ a \in 0 .. N^*_A - 1 \\ \quad Б_{розп.\Sigma} \leftarrow Б^*_{заг.} + Б^*_{CRC} \\ \quad \mathbf{for}\ \ s \in 0 .. S^{*\prime}_{cc} - 1 \\ \quad\quad n \leftarrow Б_{розп.,s} \\ \quad\quad \mathbf{BitAlloc} \leftarrow \mathbf{subvec}\left(\mathbf{ASK}_a,\ Б_{розп.\Sigma},\ Б_{розп.\Sigma} + n - 1\right) \\ \quad\quad Q^*_{s,a} \leftarrow \mathbf{b2d}(\mathbf{BitAlloc}) \\ \quad\quad Б_{розп.\Sigma} \leftarrow Б_{розп.\Sigma} + n \\ Q^* \end{array} \right. \qquad \text{(М.156)}$$

Програмний модуль (М.157) повертає масиви переданих десяткових кодів ІВМК ($\mathbf{ІВМК}^*$) і кількості виділених для них бітів ($Б^*_{ІВМК}$). Розмірність масиву: $S^{*\prime}_{cc} \times N^*_A$.

$$\begin{pmatrix} \text{ІВМК}^* \\ \text{Б}^*_{\text{ІВМК}} \end{pmatrix} := \left| \begin{array}{l} \text{for } a \in 0..\ N^*_A - 1 \\ \quad \text{Б}_{\text{ІВМКΣ}} \leftarrow \text{Б}^*_{\text{заг.}} + \text{Б}^*_{\text{CRC}} + \text{Б}^*_{\text{розп.}} \\ \quad \text{for } s \in 0..\ S''^*_{cc} - 1 \\ \quad\quad \text{if } Q^*_{s,a} = 0 \\ \quad\quad\quad \text{ІВМК}^*_{s,a} \leftarrow \text{"-"},\ \ \text{Б}^*_{\text{ІВМК}_{s,a}} \leftarrow 0 \\ \quad\quad \text{if } Q^*_{s,a} \neq 0 \\ \quad\quad\quad \text{ScFInf} \leftarrow \text{subvec}\left(\text{ASK}_a, \text{Б}_{\text{ІВМКΣ}}, \text{Б}_{\text{ІВМКΣ}} + 1\right) \\ \quad\quad\quad \text{ІВМК}^*_{s,a} \leftarrow \text{b2d}(\text{ScFInf}),\ \text{Б}^*_{\text{ІВМК}_{s,a}} \leftarrow 2 \\ \quad\quad\quad \text{Б}_{\text{ІВМКΣ}} \leftarrow \text{Б}_{\text{ІВМКΣ}} + 2 \\ \begin{pmatrix} \text{ІВМК}^* \\ \text{Б}^*_{\text{ІВМК}} \end{pmatrix} \end{array} \right. \qquad \text{(М.157)}$$

Програмний модуль (М.158) повертає масиви $S''^*_{cc} \times N^*_A$ переданих десяткових ІМК (**ІМКпер***) і кількості виділених для них бітів ($\text{Б}^*_{\text{ІМК}}$).

$$\begin{pmatrix} \text{ІМКпер}^* \\ \text{Б}^*_{\text{ІМК}} \end{pmatrix} := \left| \begin{array}{l} \text{for } a \in 0..\ N^*_A - 1 \\ \quad \text{Б}_{\text{ІМКΣ}} \leftarrow \text{Б}^*_{\text{заг.}} + \text{Б}^*_{\text{CRC}} + \text{Б}^*_{\text{розп.}} + \sum_{s=0}^{S''^*_{cc}-1} \text{Б}^*_{\text{ІВМК}_{s,a}} \\ \quad \text{for } s \in 0..\ S''^*_{cc} - 1 \\ \quad\quad \text{if } Q^*_{s,a} = 0 \\ \quad\quad\quad \text{ІМКпер}^*_{s,a} \leftarrow \text{"-"},\ \ \text{Б}^*_{\text{ІМК}_{s,a}} \leftarrow 0 \\ \quad\quad \text{if } Q^*_{s,a} \neq 0 \\ \quad\quad\quad \text{N\_ІМК} \leftarrow 3 \ \text{ if } \ \text{ІВМК}^*_{s,a} = 0 \\ \quad\quad\quad \text{N\_ІМК} \leftarrow 2 \ \text{ if } \ \text{ІВМК}^*_{s,a} = 1 \vee \text{ІВМК}^*_{s,a} = 3 \\ \quad\quad\quad \text{N\_ІМК} \leftarrow 1 \ \text{ if } \ \text{ІВМК}^*_{s,a} = 2 \\ \quad\quad\quad \mu \leftarrow 0 \\ \quad\quad\quad \text{for } \kappa \in 0..\ \text{N\_ІМК} - 1 \\ \quad\quad\quad\quad \text{index} \leftarrow \text{subvec}\left(\text{ASK}_a, \text{Б}_{\text{ІМКΣ}}, \text{Б}_{\text{ІМКΣ}} + 5\right) \\ \quad\quad\quad\quad \mu_\kappa \leftarrow \text{b2d}(\text{index}) \\ \quad\quad\quad\quad \text{Б}_{\text{ІМКΣ}} \leftarrow \text{Б}_{\text{ІМКΣ}} + 6 \\ \quad\quad\quad \text{ІМКпер}^*_{s,a} \leftarrow \mu \\ \quad\quad\quad \text{Б}^*_{\text{ІМК}_{s,a}} \leftarrow 6 \cdot \text{N\_ІМК} \\ \begin{pmatrix} \text{ІМКпер}^* \\ \text{Б}^*_{\text{ІМК}} \end{pmatrix} \end{array} \right. \qquad \text{(М.158)}$$

Програмний модуль (М.159) повертає масиви $S^{*'}_{cc} \times N^{*}_{A}$ кількості кодованих відліків або гранул (G^{*}), кількості виділених для цього бітів ($Б^{*}_{відл.}$), а також допоміжних міток, що вказують на характер кодування (F^{*}).

$$\begin{pmatrix} G^{*} \\ Б^{*}_{відл.} \\ F^{*} \end{pmatrix} := \left| \begin{array}{l} \text{for } a \in 0..N^{*}_{A}-1 \\ \quad \text{for } s \in 0..S^{*'}_{cc}-1 \\ \quad\quad q \leftarrow Q^{*}_{s,a},\ S \leftarrow Nstep_{s,q} \\ \quad\quad Б^{*}_{відл.s,a} \leftarrow 0, G^{*}_{s,a} \leftarrow 0, F^{*}_{s,a} \leftarrow 0 \text{ if } S = 0 \\ \quad\quad Б^{*}_{відл.s,a} \leftarrow 5, G^{*}_{s,a} \leftarrow 12, F^{*}_{s,a} \leftarrow \text{'гранула' if } S = 3 \\ \quad\quad Б^{*}_{відл.s,a} \leftarrow 7, G^{*}_{s,a} \leftarrow 12, F^{*}_{s,a} \leftarrow \text{'гранула' if } S = 5 \\ \quad\quad Б^{*}_{відл.s,a} \leftarrow 10, G^{*}_{s,a} \leftarrow 12, F^{*}_{s,a} \leftarrow \text{'гранула' if } S = 9 \\ \quad\quad \text{if } S \neq 0 \wedge S \neq 3 \wedge S \neq 5 \wedge S \neq 9 \\ \quad\quad\quad Б^{*}_{відл.s,a} \leftarrow \log_2(S+1),\ G^{*}_{s,a} \leftarrow 36 \\ \quad\quad\quad F^{*}_{s,a} \leftarrow \text{'окремо'} \\ \begin{pmatrix} G^{*} \\ Б^{*}_{відл.} \\ F^{*} \end{pmatrix} \end{array} \right. \qquad \text{(М.159)}$$

Кінцевим етапом підготовки до видобування бітів повідомлення є одержання з аудіокадрів **ASK** відлікових контейнерів. Програмний модуль, що формує масив двійкових слів кодованих субсмугових відліків, — (М.160). Результуючий масив **Vq2*** є тривимірним: $S^{*'}_{cc} \times N^{*}_{A} \times \max(b)$, і, за умови відсутності спотворень контенту потоку стиснутого аудіосигналу, — повністю відповідним масиву **Vq2**, отриманому по виконанню програмного модуля (М.145).

Зчитування найменших значущих бітів

У випадку приховання повідомлень за алгоритмом заміни НЗБ КК субсмугових відліків або гранул потоків аудіокадрів *MPEG Layer II*, зчитування НЗБ реалізується програмним модулем (М.161). При цьому для спрощення передбачається, що видобуване повідомлення має фіксовану, відому приймальній стороні довжину L^{*}_{M} (у випадку використання повідомлень непостійної довжини, на передавальному боці перед етапом вбудовування повідомлення можна попередньо помістити в нього характерні таємні мітки, що визначатимуть його початок і кінець). Зчитуються НЗБ кодів тільки тих відліків і гранул субсмуг аудіокадрів, що за результатами ДРБ були кодовані кількістю бітів, яка перевищує поріг λ (див. (5.59) і (М.146)).

$$
\begin{aligned}
Vq2^* := \;& Vq2^* \leftarrow 0 \cdot Q^* \\
& \text{for } a \in 0..N^*_A - 1 \\
& \quad Б_{відлΣ} \leftarrow Б^*_{заг.} + Б^*_{CRC} + Б^*_{розп.} + \sum_{s=0}^{S^{**}_{CC}-1} Б^*_{ІВМК_{s,a}} + \sum_{s=0}^{S^{**}_{CC}-1} Б^*_{ІМК_{s,a}} \\
& \quad b \leftarrow 0 \\
& \quad \text{while } b < 34 \\
& \quad\quad \text{for } s \in 0..S^{**}_{CC} - 1 \\
& \quad\quad\quad \text{if } F^*_{s,a} = \text{'окремо'} \\
& \quad\quad\quad\quad \text{for } j \in 0..2 \\
& \quad\quad\quad\quad\quad \left| Vq2^*_{s,a} \right|_{j+b} \leftarrow \text{subvec}\left(ASK_a, Б_{відлΣ}, Б_{відлΣ} + Б^*_{відл._{s,a}} - 1\right) \\
& \quad\quad\quad\quad\quad Б_{відлΣ} \leftarrow Б_{відлΣ} + Б^*_{відл._{s,a}} \\
& \quad\quad\quad \text{if } F^*_{s,a} = \text{'гранула'} \\
& \quad\quad\quad\quad \left| Vq2^*_{s,a} \right|_b \leftarrow \text{subvec}\left(ASK_a, Б_{відлΣ}, Б_{відлΣ} + Б^*_{відл._{s,a}} - 1\right) \\
& \quad\quad\quad\quad Б_{відлΣ} \leftarrow Б_{відлΣ} + Б^*_{відл._{s,a}} \\
& \quad\quad\quad Vq2^*_{s,a} \leftarrow \text{'-'} \text{ if } F^*_{s,a} = 0 \\
& \quad\quad\quad b \leftarrow b + 3 \text{ if } s = S^{**}_{CC} - 1 \\
& Vq2^*
\end{aligned}
\qquad (M.160)
$$

По завершенні видобування одержується вектор бінаризованого повідомлення **Mbin***, яке для можливості подальшого відтворення відповідними засобами має бути декодоване у належному декодері. Якщо, наприклад, відомо, що кінцевий формат повідомлення — це текст, то декодером може виступати прикінцевий сегмент програмного модуля (М.161).

Визначення співвідношення між субсмуговими відліками

У разі приховання повідомлень класу ЦВЗ (див. (5.60) і (М.147)), для можливості здійснення порівняльних оцінок значень реквантованих субсмугових відліків $\mathbf{X\Sigma}^*$ треба виконати попередню процедуру реквантування (див. пункт А.11.4 Додатка), що реалізується за допомогою програмного модуля (М.162).

Безпосередньо видобування бітів повідомлення-ЦВЗ виконує програмний модуль (М.163). При цьому для спрощення передбачається, що бітова довжина видобуваного ЦВЗ дорівнює кількості аудіокадрів (опис дій сторін стеганографічного обміну у випадку змінної довжини ЦВЗ можуть бути знайдені, наприклад, у працях [3,112]).

```
M* :=  μ ← 0
       for  a ∈ 0.. N*A − 1
         for  s ∈ 0.. S'*cc − 1
           for  b ∈ 0.. 35
             break  if  μ = L*M
             if  Б*відл.s,a > λ
               if  F*s,a = 'окремо'
                 Mbin*μ ← [|Vq2*s,a|b]last[|Vq2*s,a|b]
                 μ ← μ + 1
               if  F*s,a = 'гранула' ∧ mod(b,3) = 0
                 Mbin*μ ← [|Vq2*s,a|b]last[|Vq2*s,a|b]
                 μ ← μ + 1
       for  m ∈ 0.. 0.125·rows(Mbin*) − 1
         Mvec*m ← b2d[subvec[Mbin*, 8·m, 8·(m+1)−1]]
         if  Mvec*m < 32 ∧ Mvec*m ≠ 10 ∧ Mvec*m ≠ 13
           Mvec*m ← 32
       vec2str(Mvec*)
```
(M.161)

Крім того, для врахування спотворень кодових комбінацій відліків, що могли виникнути в результаті атаки повторним стисканням із втратами, рішення про кожний переданий біт ЦВЗ приймається в (M.163) шляхом аналізу співвідношень усереднених на множині з 36 відліків псевдовипадково обраних (за узгодженим з передавальною стороною алгоритмом) субсмуг **s1** і **s2** реквантованих відліків поточного аудіокадру.

На проміжному етапі результатом видобування є вектор бінаризованого ЦВЗ **Wbin***, що для потреб подальшого відтворення належними засобами має бути декодований у відповідному декодері системи кінцевої обробки (див. рис. 2.3). Якщо відомо, наприклад, що кінцевий формат ЦВЗ повинен бути графічним і таким, що має певні геометричні параметри (піксельні розміри по висоті та ширині), то декодером можна вважати прикінцевий сегмент програмного модуля (M.163).

$$
\begin{aligned}
&X\Sigma^* := \left|\begin{array}{l}
X\Sigma^* \leftarrow 0 \cdot Q^* \\
\text{for } a \in 0..N^*_A - 1 \\
\quad \text{for } s \in 0..S^*_{cc} - 1 \\
\quad\quad q \leftarrow Q^*_{s,a},\ S \leftarrow Nstep_{s,q},\ n \leftarrow Б^*_{відл.s,a} \\
\quad\quad step\_ind \leftarrow пошук(S, Step)_0 - 1 \\
\quad\quad \text{if } F^*_{s,a} = \text{'окремо'} \\
\quad\quad\quad \text{for } b \in 0..35 \\
\quad\quad\quad\quad bin \leftarrow |Vq2^*_{s,a}|_b,\ bin_0 \leftarrow bin_0 \oplus 1,\ знак \leftarrow -bin_0 \\
\quad\quad\quad\quad bin \leftarrow if(знак = -1, compl\_code2(bin, n), bin),\ dec \leftarrow b2d(bin) \\
\quad\quad\quad\quad зсув \leftarrow 2^{n-1} \cdot 0.999999999999,\ \Xi_b \leftarrow знак \cdot dec/зсув \\
\quad\quad\quad\quad X\Sigma_b \leftarrow C_{step\_ind} \cdot (\Xi_b + D_{step\_ind}) \\
\quad\quad if\ F^*_{s,a} = \text{'гранула'} \\
\quad\quad\quad b \leftarrow 0 \\
\quad\quad\quad while\ b < 34 \\
\quad\quad\quad\quad bin\_v \leftarrow |Vq2^*_{s,a}|_b,\ v \leftarrow b2d(bin\_v) \\
\quad\quad\quad\quad for\ i \in 0..2 \\
\quad\quad\quad\quad\quad ocm_i \leftarrow mod(v, S),\ v \leftarrow trunc(v/S) \\
\quad\quad\quad\quad x \leftarrow ocm_0, y \leftarrow ocm_1, z \leftarrow ocm_2, \nu \leftarrow if(S = 3, 2, if(S = 5, 3, 4)) \\
\quad\quad\quad\quad bin\_x \leftarrow d2b(x, \nu),\ bin\_y \leftarrow d2b(y, \nu),\ bin\_z \leftarrow d2b(z, \nu) \\
\quad\quad\quad\quad bin\_x_0 \leftarrow bin\_x_0 \oplus 1,\ bin\_y_0 \leftarrow bin\_y_0 \oplus 1,\ bin\_z_0 \leftarrow bin\_z_0 \oplus 1 \\
\quad\quad\quad\quad знак\_x \leftarrow -bin\_x_0,\ знак\_y \leftarrow -bin\_y_0,\ знак\_z \leftarrow -bin\_z_0 \\
\quad\quad\quad\quad bin\_x \leftarrow if(знак\_x = -1, compl\_code2(bin\_x, \nu), bin\_x) \\
\quad\quad\quad\quad dec\_x \leftarrow b2d(bin\_x) \\
\quad\quad\quad\quad bin\_y \leftarrow if(знак\_y = -1, compl\_code2(bin\_y, \nu), bin\_y) \\
\quad\quad\quad\quad dec\_y \leftarrow b2d(bin\_y) \\
\quad\quad\quad\quad bin\_z \leftarrow if(знак\_z = -1, compl\_code2(bin\_z, \nu), bin\_z) \\
\quad\quad\quad\quad dec\_z \leftarrow b2d(bin\_z) \\
\quad\quad\quad\quad зсув \leftarrow if(S = 3, 1.999999999999, if(S = 5, 3.999999999999, 7.999999999999)) \\
\quad\quad\quad\quad \Xi\_x \leftarrow знак\_x \cdot \dfrac{dec\_x}{зсув},\ \Xi\_y \leftarrow знак\_y \cdot \dfrac{dec\_y}{зсув},\ \Xi\_z \leftarrow знак\_z \cdot \dfrac{dec\_z}{зсув} \\
\quad\quad\quad\quad \Xi_b \leftarrow \Xi\_x,\ \Xi_{b+1} \leftarrow \Xi\_y,\ \Xi_{b+2} \leftarrow \Xi\_z \\
\quad\quad\quad\quad X\Sigma_b \leftarrow C_{step\_ind} \cdot (\Xi\_x + D_{step\_ind}) \\
\quad\quad\quad\quad X\Sigma_{b+1} \leftarrow C_{step\_ind} \cdot (\Xi\_y + D_{step\_ind}) \\
\quad\quad\quad\quad X\Sigma_{b+2} \leftarrow C_{step\_ind} \cdot (\Xi\_z + D_{step\_ind}) \\
\quad\quad\quad\quad b \leftarrow b + 3 \\
\quad\quad X\Sigma^*_{s,a} \leftarrow X\Sigma\ \text{ if }\ F^*_{s,a} \neq 0 \\
X\Sigma^*
\end{array}\right.
\end{aligned}
$$

(M.162)

```
W* := | for a ∈ 0.. N*_A − 1                                                    (M.163
      |   s1 ← K1_a , s2 ← K2_a
      |   Wbin*_a ← round(rnd(1)) on error Wbin*_a ← mean(XΣ*_s1,a) < mean(XΣ*_s2,a)
      | W* ← subvec(Wbin*, 0, 7)
      | for i ∈ 1.. rows(Wbin*)/8 − 1
      |     W* ← augment[W*, subvec[Wbin*, i·8, (i+1)·8 − 1]]
      | W*·255
```

Зчитування найменш значущих бітів у псевдовипадково обраних субсмугових відліках

Процедуру зчитування НЗБ кодових комбінацій незалежно кодованих псевдовипадково обраних у межах *MPEG*-аудіокадрів субсмугових відліків реалізує програмний модуль (М.164).

```
M* := | Δμ ← 0                                                                (M.164
      | for a ∈ 0.. N*_A − 1
      |   for s ∈ 0.. S'*_CC − 1
      |     for b ∈ 0.. 35
      |       if F*_s,a = 'окремо' ∧ s ≥ ζ
      |         μ ← пошук(b, β)_0 + Δμ if b < rows(β)
      |         Mbin*_μ ← [Vq2*_s,a]_b  last[Vq2*_s,a]_b
      |         Δμ ← Δμ + rows(β) − ζ if Б*_відл._s,a > λ ∧ b = 35
      | for m ∈ 0.. rows(Mbin*)/8 − 1
      |   Mvec*_m ← b2d[subvec[Mbin*, 8·m, 8·(m+1) − 1]]
      |   Mvec*_m ← 32 if Mvec*_m < 32 ∧ Mvec*_m ≠ 10 ∧ Mvec*_m ≠ 13
      | vec2str(Mvec*)
```

При цьому вважається, що для стеганографічного обміну використовуються генератори ПВП, що для обраних ключів $k \in \mathbf{K}$ виробляють ідентичні послідовності індексів β, а також узгоджені пороги розрядності кодування відліків-контейнерів (λ) і субсмуг, що містять відібрані відлікові контейнери (ζ).

Одержаний по завершенні видобування вектор бінаризованого повідомлення **Mbin*** перед подальшим відтворенням за допомогою відповідних засобів декодується у декодері системи кінцевої обробки (див. рис. 2.3). Наприклад, якщо кінцевий формат повідомлення має бути текстовим, функції декодера виконуються останнім сегментом програмного модуля (М.164).

5.5. Приховування даних у тексті

Для приховування конфіденційних повідомлень у тексті (або так звана *лінгвістична стеганографія*) використовується або спричинена надлишковістю мови надлишковість сукупності текстових знаків, або спеціальні формати представлення останніх.

За багатьма причинами оцифрована версія тексту є найскладнішим і незручним місцем для приховування даних — на відміну від своєї «жорсткої» копії (наприклад, паперової), що може бути оброблена як високоструктуроване зображення і, як наслідок, є такою, що легко піддається різноманітним методам обробки на зразок незначних змін формату текстових зразків, регулювання відстані між певними парами символів (кернинг), відстані між рядками тощо. В значній мірі це викликане відносним дефіцитом у текстовому файлі надлишкової інформації, особливо в порівнянні з графічними чи аудіо даними. У той час, як до зображення/звуку у більшості випадків виявляється можливим внести непомітні/невідчутні модифікації, навіть додаткова літера або крапка в тексті можуть бути легко помічені випадковим читачем.

Таким чином, приховування даних у текстових послідовностях вимагає пошуку таких їхніх модифікацій, які були б непомітними переважною більшістю читачів. Автори [14] розглядають три групи методів, що зазнали найбільшого поширення при вбудовуванні приховуваних даних до тексту:

- *методи довільного інтервалу*, що здійснюють вбудовування шляхом маніпуляції з пробільними символами (вільним місцем на видрукованій полосі);
- *синтаксичні методи*, що працюють з пунктуацією;
- *семантичні методи*, до основи яких покладене залежне від приховуваних бітів даних маніпулювання словами.

5.5.1. Методи довільного інтервалу

Існує, щонайменше, дві причини, за якими маніпулювання вільним місцем у тексті за певних випадків здатне показати досить непогані результати. По-перше, зміна кількості пробілів наприкінці текстового рядка абсолютно не спричиняє істотних змін у значенні фрази або речення. По-друге, незначні модифікації вільного місця пересічний читач навряд чи навіть і помітить.

У [14] запропоновано три методи, що для приховування даних використовують вільне місце в тексті. Ці методи оперують з інтервалами між реченнями, пропусками в кінці текстових рядків, а також інтервалами між словами в тексті, що був вирівняний по ширині.

5.5.1.1. Метод зміни інтервалу між реченнями

Зазначений метод дозволяє вбудовувати до тексту повідомлення, що має двійковий формат, шляхом розміщення одного чи двох пробілів після кожного символу завершення речення (наприклад, крапки у звичайному тексті або крапки з комою для коду програм на мові *C++* тощо). При цьому, наприклад, одиночним пробілом кодується біт «1», подвійним — біт «0». Проте, такий, без сумніву, простий метод має й низку недоліків. По-перше, він є неефективним, оскільки для вбудовування незначної кількості бітів вимагає наявності текстового контейнера великого обсягу (зокрема, один біт, прихований в одному реченні, є еквівалентним швидкості передавання близько 1 біта даних на 140 байтів текстового контейнера — за умови, що в середньому речення становить собою 2 рядки по 70 символів кожен). По-друге, можливість приховування залежить від структури тексту-контейнера (деякі тексти, як наприклад, верлібри або вільні вірші характеризуються відсутністю стійких узгоджених або однозначних знаків завершення рядка). По-третє, деякі з текстових редакторів автоматично встановлюють після крапки наприкінці речення лише один пробіл. Зрештою ж, як зазначається у [14], непослідовне використання вільних місць є досить помітним для читача.

Наведемо приклад реалізації даного методу в середовищі *Mathcad*.

Крок 1

Нехай в якості контейнера використовується текст, візуалізований фрагмент якого наведено на рис. 5.147, **C** := **READBIN**("C.txt", "byte"), а приховуване повідомлення має наступний зміст:

M := "© Alex Puzyrenko, 2017".

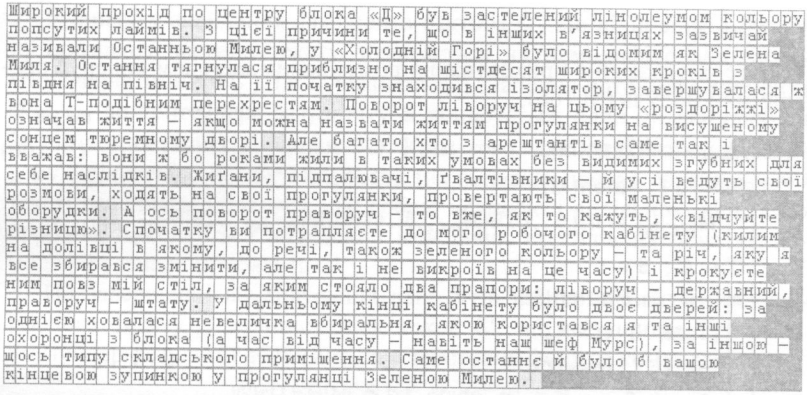

Широкий прохід по центру блока «Д» був застелений лінолеумом кольору попсутих лаймів. З цієї причини те, що в інших в'язницях зазвичай називали останньою Милею, у «Холодній Горі» було відомим як Зелена Миля. Остання тягнулася приблизно на шістдесят широких кроків з півдня на північ. На її початку знаходився ізолятор, завершувалася ж вона Т-подібним перехрестям. Поворот ліворуч на цьому «роздоріжжі» означав життя — якщо можна назвати життям прогулянки на висущеному сонцем тюремному дворі. Але багато хто з арештантів саме так і вважав: вони ж бо роками жили в таких умовах без видимих згубних для себе наслідків. Жиґани, підпалювачі, ґвалтівники — й усі ведуть свої розмови, ходять на свої прогулянки, провертають свої маленькі оборудки. А ось поворот праворуч — то вже, як то кажуть, «відчуйте різницю». Спочатку ви потрапляєте до мого робочого кабінету (килим на долівці в якому, до речі, також зеленого кольору — та річ, яку я все збирався змінити, але так і не викроїв на це часу) і крокуєте ним повз мій стіл, за яким стояло два прапори: ліворуч — державний, праворуч — штату. У дальньому кінці кабінету було двоє дверей: за одними ховалася невеличка вбиральня, якою користався я та інші охоронці з блока (а час від часу — навіть наш шеф Мурс), за іншою — щось типу складського приміщення. Саме останнє було б вашою кінцевою зупинкою у прогулянці Зеленою Милею.

Рис. 5.147. Фрагмент оригіналу тексту-контейнера[XXVIII]

Крок 2

Текстовою змінною π := ". " (крапка + пробіл) позначимо той фрагмент контейнера, який сигналізуватиме про закінчення речень. У *ASCII*-кодуванні дана змінна являтиме собою вектор $\mathbf{str2vec}(\pi)^T = (46\ 32)$.

Перевірку достатності кількості речень у тексті для приховування заданої кількості бітів повідомлення

$$L_M := 8 \cdot \mathbf{strlen}(M) = 176 \text{ бітів}$$

виконаємо за допомогою програмного модуля (М.165), що підраховує наявну кількість елементів π в обраному в якості контейнера тексті.

$$N_\pi := \left| \begin{array}{l} N_\pi \leftarrow 0 \\ \text{for } i \in 1..\mathbf{rows}(C) - 1 \\ \quad \left| \begin{array}{l} \Pi \leftarrow \mathbf{subvec}(C, i, i+1) \\ N_\pi \leftarrow N_\pi + 1 \quad \text{if } \Pi = \mathbf{str2vec}(\pi) \end{array} \right. \\ N_\pi \end{array} \right. \qquad \text{(М.165)}$$

Для представленого на рис. 5.147 фрагмента, наприклад, $N_\pi = 11$, що буде достатнім для приховання тільки одного (приміром, першого) символу повідомлення (8 бітів).

Отже, в зображеному на рис. 5.147 фрагменті тексту розміром $\mathbf{rows}(C) = 1394$ байти (11152 біти) можна приховати лише 11 бітів повідомлення.

[XXVIII] Використано уривок з роману С. Кінга «Зелена миля» (1996), пер. з англ. О. Пузиренка (Київ : «Гуртом», 2017. — 704 с.).

Крок 3

Стеганографічне приховування розгляданим методом виконує програмний модуль (М.166).

```
S := | M_vec ← str2vec(M)                                                    (M.166)
     | for j ∈ 1..strlen(M)
     |   M_vec_bin ← if(j = 1, D2B(M_vec_1, 8), stack(M_vec_bin, D2B(M_vec_j, 8)))
     | C' ← C , S ← 0
     | for μ ∈ 1..8·strlen(M)
     |   break if μ > N_π
     |   for i ∈ 1..rows(C')
     |     S ← if[S ≠ 0, stack[S, (C'_i)], (C'_i)]
     |     if i < rows(C')
     |       Π ← subvec(C', i, i + 1) , Π' ← str2vec(π)
     |       if Π = Π'
     |         if M_vec_bin_μ = 1
     |           C' ← (13) on error C' ← subvec(C', i + 2, rows(C'))
     |           S ← stack[S, (32)]
     |           while C'_1 = 32
     |             C' ← subvec(C', 2, rows(C'))
     |           break
     |         if M_vec_bin_μ = 0
     |           C' ← (13) on error C' ← subvec(C', i + 2, rows(C'))
     |           S ← stack[S, (32 ; 32)]
     |           while C'_1 = 32
     |             C' ← subvec(C', 2, rows(C'))
     |           break
     | stack(S, C')
```

На початку модуля рядок символів **M** перетворюється на вектор двійкових даних $\mathbf{M}_{\mathbf{vec\_bin}}$. Для вбудовування кожного біта $\mathbf{M}_{\mathbf{vec\_bin}}$ (у кількості, що обмежена значенням $\mathbf{N}_{\pi}$) проводиться пошук кінця наступного речення (елемента π) у масиві представленого в *ASCII*-кодах тексту **C'**, який ще не зазнав модифікації вбудовуваними бітами. Паралельно формується контейнер-результат **S** — шляхом дописування в його кінець **i**-го елемента вектора **C'**.

У випадку знайдення фрагмента вектора **C'**, що відповідає вектору **π**, подальша частина тексту зберігається як немодифікована (перевизначення змінної **C'**), і в залежності від значення поточного елемента $\mathbf{M}_{\mathbf{vec\_bin}}$ до контейнера-результату **S** дописується один чи два пробіли (*ASCII*-код останніх дорівнює 32). Крім того, для запобігання можливому виникненню неоднозначності в тих випадках, коли в оригінальному тексті речення відділялися між собою крапкою і довільною кількістю пробілів, виконується видалення можливих пробілів на початку немодифікованого фрагмента **C'**.

По завершенні вбудовування всіх бітів повідомлення, до результуючого тексту **S** приписується текст **C'**, що залишився без змін. Контейнер-результат зберігається у текстовий файл за командою:

WRITEBIN("S.txt", "byte", 0) := **S**.

Результат приховання першого символу повідомлення **M** (символу "©"), *ASCII*-код якого у двійковій формі представлення: $\mathbf{D2B}(\mathbf{str2vec}(\mathbf{M})_1, 8)^{\mathrm{T}} = (1_{\mathrm{LSb}}\,0\,0\,1\,0\,1\,0\,1_{\mathrm{MSb}})$, можна побачити на рис. 5.148.

Широкий прохід по центру блока «Д» був застелений лінолеумом кольору попсутих лаймів. З цієї причини те, що в інших в'язницях називали Останньою Милею, у «Холодній Горі» було відоме як Зелена Миля. Остання тягнулася приблизно на шістдесят широких кроків з півдня на північ. На її початку знаходився ізолятор, завершувалася ж вона Т-подібним перехрестям. Поворот ліворуч на цьому «роздоріжжі» означав життя – якщо можна назвати життям прогулянки на висушеному сонцем тремтячому дворі. Але багато хто з арештантів саме так і вважав: вони ж бо роками жили в таких умовах без видимих згубних для себе наслідків. Жиґани, підпалювачі, ґвалтівники – и усі ведуть свої розмови, ходять на свої прогулянки, проверчають свої маленькі оборудки. А ось поворот праворуч – то вже, як то кажуть, «відчуйте різницю». Спочатку ви потрапляєте до мого робочого кабінету (килим на долівці в якому, до речі, також зеленого кольору – та річ, якну я все збирався змінити, але так і не викроїв на це часу) і кроките ним повз мій стіл, за яким стояло два прапори: ліворуч – державний, праворуч – штату. У дальньому кінці кабінету було двоє дверей: за одні єю ховалася невеличка вбиральня, якою користався я та інші охоронці з блока (а час від часу – навіть наш шеф Мурс), за іншою – щось типу складського приміщення. Саме останнє й було б вашою кінцевою зупинкою у прогулянці Зеленою Милею.

Рис. 5.148. Фрагмент текстового контейнера-результату, заповненого методом зміни інтервалу між реченнями

Крок 4

Видобування прихованої інформації проводиться за допомогою програмного модуля (М.167) при $\mathbf{S}^* := \mathbf{S}$, $\pi^* := \pi$.

Алгоритм дій наступний: виконується пошук кінця речення (елемент π^*) у представленому в *ASCII*-кодах тексті $\mathbf{S}^*$. У випадку знаходження, аналізується кількість пробілів після крапки, на підставі чого робиться висновок про значення поточного видобутого біта повідомлення.

Наприкінці модуля вектор двійкових даних перетворюється на вектор десяткових *ASCII*-кодів **M*vec** і, в підсумку, на текстовий рядок символів **M*** — копію прихованого повідомлення.

$$
\begin{aligned}
&\mathbf{M}^* := \left|
\begin{array}{l}
\mu \leftarrow 1 \quad \Pi^* \leftarrow \mathbf{str2vec}(\pi^*) \\
\textbf{for}\ \ i \in 1..\mathbf{rows}(\mathbf{S}^*) - 2 \\
\quad \left|
\begin{array}{l}
\Pi \leftarrow \mathbf{subvec}(\mathbf{S}^*, i, i+1) \quad \textbf{if}\ \ i < \mathbf{rows}(\mathbf{S}^*) \\
\textbf{if}\ \ \Pi = \Pi^* \\
\quad \left|
\begin{array}{l}
\mathbf{M}^*\mathbf{bin}_{\mu} \leftarrow \textbf{if}\left(\mathbf{S}^*_{i+1} = \mathbf{S}^*_{i+2} = 32, 0, 1\right) \\
\mu \leftarrow \mu + 1
\end{array}
\right.
\end{array}
\right. \\
\textbf{for}\ \ j \in 1..\dfrac{\mathbf{rows}(\mathbf{M}^*\mathbf{bin})}{8} \\
\quad \left|
\begin{array}{l}
\mathbf{M}^*\mathbf{vec}_j \leftarrow \mathbf{B2D}(\mathbf{subvec}(\mathbf{M}^*\mathbf{bin}, 8\cdot j - 7, 8\cdot j)) \\
\mathbf{M}^*\mathbf{vec}_j \leftarrow 32 \quad \textbf{if}\ \ \mathbf{M}^*\mathbf{vec}_j < 32 \wedge \mathbf{M}^*\mathbf{vec}_j \neq 10 \wedge \mathbf{M}^*\mathbf{vec}_j \neq 13
\end{array}
\right. \\
\mathbf{vec2str}(\mathbf{M}^*\mathbf{vec})
\end{array}
\right.
\end{aligned}
\qquad \text{(M.167)}
$$

5.5.1.2. Метод зміни кількості чи типу пробілів у кінці текстових рядків

Ще один метод використання вільних місць текстових полос для вбудовування конфіденційних даних полягає у додаванні пробілів в кінець кожного текстового рядка. Кількість додаваних пробілів при цьому залежить від біта, що наразі вбудовується. Додавання одного пробілу здатне закодувати один біт на рядок (наприклад, відсутність пробілу наприкінці рядка — це «0», наявність — «1»); до трьох пробілів — два біти; до семи пробілів — три біти і т. д., істотно збільшуючи, порівняно з попереднім методом, кількість інформації, яку виникає можливість приховати в рівному за обсягом тексті.

Додаткові переваги даного методу полягають у тому, що він може бути застосовний до будь-якого тексту; зміни в форматі є досить непомітними, оскільки вільні місця, що при цьому використовуються, є периферійними по відношенню до основного тексту.

Недоліком даного (як, зрештою, і попереднього) методу є те, що деякі програми можуть навіть ненавмисно видаляти додатково внесені пробіли. Крім того, характерним недоліком саме даного методу є неможливість видобування прихованих даних з паперової копії тексту (через невидимість пробілів і відсутність правої границі у вигляді наступного, видимого символу).

Нами запропоновано наступний різновид реалізації даного методу у програмі *Mathcad*.

Крок 1

Нехай порожній текстовий контейнер і приховуване повідомлення — ті самі, що й у розглянутому попередньо методі.

Крок 2

Вбудовування бітів повідомлення виконаємо за допомогою програмного модуля (М.168).

```
S := Mvec ← str2vec(M)                                                    (M.168)
     for j ∈ 1..strlen(M)
       Mvec_bin ← if(j = 1, D2B(Mvec_1, 8), stack(Mvec_bin, D2B(Mvec_j, 8)))
     n ← 0, r ← 1
     for i ∈ 1..rows(C)
       Lr ← n − 1, r ← r + 1, n ← 0 if Ci = 10
       n ← n + 1 otherwise
     C' ← C, S ← 0, r ← 1, μ ← 1
     while μ ≤ 8·strlen(M)
       break if r > rows(L)
       for i ∈ 1..rows(C')
         S ← if[S ≠ 0, stack[S, (C'i)], (C'i)] if C'i ≠ 13
         if C'i = 13
           C' ← subvec(C', i+2, rows(C')) if i+2 ≤ rows(C')
           κ ← 1
           while max(L) − Lr ≥ κ
             S ← stack[S, if[Mvec_binμ, («1»), («0»)]]
             μ ← μ + 1
             break if μ > 8·strlen(M)
             κ ← κ + 1
           S ← stack[S, (13 10)]
           r ← r + 1
           break
     S ← stack(S, C') if r < rows(L)
     S
```

Після одержання вектора $\mathbf{M}_{\mathbf{vec\_bin}}$ двійкового представлення повідомлення **M**, що підлягає прихованню, виконується підрахунок кількості

символів **n** у кожному текстовому рядку **r** [XXIX]. Підрахована кількість заноситься до **r**-го елемента вектора **L**.

На самому початку масив **C'** являє собою текст, який ще не зазнав модифікації, а контейнер-результат **S** порожній. Формування стеганограми здійснюється шляхом дописування після поточного (останнього) елемента вектора **S** значення **i**-го елемента вектора **C'**, якщо той не являє собою символ повернення каретки *CR*. У протилежному випадку виконується перевизначення змінної **C'** — як такої, що містить немодифіковану частину тексту.

Кількість бітів **к**, яку можна приховати в поточному (**r**-му) рядку тексту, визначається різницею між кількістю символів у найдовшому з усіх рядків тексту (максимальне значення вектора **L**) та кількістю символів у рядку **r**. При цьому, якщо приховується біт «0», до контейнера дописується значення **«0»**:=32 (звичайний пробіл), а якщо біт «1» – значення **«1»**:=160 (нерозривний пробіл). Після вбудовування **к** бітів, рядок завершується парою символів *CR/LF*.

Після вбудовування всіх 8·**strlen**(**M**) бітів повідомлення (або ж при досягненні кінця тексту) цикл переривається. Якщо рядок, що модифікувався останнім, не є останнім у тексті (загальну кількість рядків тексту можна визначити як кількість елементів вектора **L**), контейнер-результат доповнюється немодифікованим залишком тексту **C'**.

Приклад вбудовування повідомлення **M** до контейнера **C** зображено на рис. 5.149 (для наочності нерозривний пробіл показано підкресленням).

Широкий прохід по центру блока «Д» був застелений лінолеумом кольору допустих лаймів. З цієї причини те, що в інших в'язницях зазвичай називали останньою Милею, у «Холодній Горі» було відоме як Зелена Миля. Остання тягнулася приблизно на шістдесят широких кроків з півдня на північ. На її початку знаходився ізолятор, завершувалася вона Т-подібним перехрестям. Поворот ліворуч на цьому «роздоріжжі» означав життя – якщо можна назвати життям прогулянки на відсуненому сонцезахисному дворі. Але багато хто з арештантів саме так і вважає: вони ж бо роками жили в таких умовах без видимих згубних для себе наслідків. Хулігани, підпалювачі, ґвалтівники – усі вони ведуть свої розмови, ходять на свої прогулянки, проверчують свої маленькі обурення. А ось поворот праворуч – то вже, як то кажуть, «відчуйте різницю». Спочатку ви потрапляєте до мого робочого кабінету (килим на долівці в ньому, до речі, також зеленого кольору – та річ, яку я все збираюся змінити, але так і не викроїв на це часу), і крокуєте ним повз мій стіл, за яким стоїть два прапори: ліворуч – державний, праворуч – штату. У дальньому кінці кабінету будуть дві двері: за однією ховалася невеличка убиральня, якою користувався я та інші охоронці з блока (а нас від часу – навіть наш шеф Мурс), за другою щось типу складського приміщення. Саме останнє и було б вашою кінцевою зупинкою у прогулянці Зеленою Милею.

Рис. 5.149. Фрагмент текстового контейнера-результату, заповненого методом зміни типу пробілів в кінці рядків

[XXIX] Нагадаємо, що кожен рядок тексту завершується парою службових символів повернення каретки *CR* (*ASCII*-код 13) й перенесення рядка *LF* (*ASCII*-код 10), що в звичайному режимі на екран і на друк не виводяться.

Таким чином, у зображеному на рис. 5.147 фрагменті тексту розміром **rows(C)** = 1394 байти (11152 біти) приховано 76 бітів повідомлення (майже у 7 разів більше за метод, розглянутий у підп. 5.5.1.1).

Крок 3

Видобування бітів прихованих даних з контейнера $S^* := S$ здійснюється за допомогою програмного модуля (М.169).

$$
\begin{array}{l}
M^* := \mu \leftarrow 1 \\
\quad \text{for } i \in 1..\text{rows}(S^*) \\
\quad\quad \text{if } S^*_i = 13 \\
\quad\quad\quad j \leftarrow i \\
\quad\quad\quad \text{while } S^*_{j-1} = \text{«0»} \vee S^*_{j-1} = \text{«1»} \\
\quad\quad\quad\quad j \leftarrow j - 1 \\
\quad\quad\quad \text{for } m \in j..i-1 \qquad\qquad \text{if } j \neq i \\
\quad\quad\quad\quad M^*\text{bin}_\mu \leftarrow \text{if}(S^*_m = \text{«0»}, 0, 1) \\
\quad\quad\quad\quad \mu \leftarrow \mu + 1 \\
\quad \text{for } j \in 1..\dfrac{\text{rows}(M^*\text{bin})}{8} \\
\quad\quad M^*\text{vec}_j \leftarrow \text{B2D}(\text{subvec}(M^*\text{bin}, 8\cdot j - 7, 8\cdot j)) \\
\quad\quad M^*\text{vec}_j \leftarrow 32 \quad \text{if } M^*\text{vec}_j < 32 \wedge M^*\text{vec}_j \neq 10 \wedge M^*\text{vec}_j \neq 13 \\
\quad \text{vec2str}(M^*\text{vec})
\end{array}
\qquad \text{(М.169)}
$$

При цьому по всій довжині вектора S^* ведеться пошук символів *CR* (*ASCII*-код 13). У випадку знайдення проводиться аналіз символів, що розміщені до нього, на предмет відповідності їхніх кодів кодам звичайного («0»=32) і нерозривного («1»=160) пробілів. На підставі отриманих результатів формується вектор двійкових даних M^*_{bin}, який, у свою чергу, згодом конвертується на текстовий рядок символів M^*.

5.5.1.3. Метод зміни кількості пробілів між словами вирівняного по ширині тексту

Даний метод дозволяє приховувати дані у вільних місцях тексту, вирівняного по ширині. При цьому дані вбудовуються шляхом керованого обрання позицій, в яких буде розміщено додаткові пробіли. Один пробіл між словами інтерпретується як «0». Два пробіли — як «1». У середньому метод дозволяє вбудувати по декілька бітів до одного рядка.

Через обмеження, що накладаються вирівнюванням тексту по ширині, не кожен пробіл між словами може використовуватися для вбудовування даних. Для можливості прийняття приймальною стороною однозначного рішення при визначенні, які ж саме з пробілів між словами приховують вбудовану інформацію, а які є частиною оригінального тексту, у [14] запропоновано використовувати метод вбудовування, подібний до манчестерського коду.

В результаті такого кодування біти групуються попарно, причому послідовність «01» інтерпретується як «1», «10» — як «0», а пари «00» і «11» вважаються порожніми. Наприклад, видобуте повідомлення «1010 0111» у підсумку зводиться до «001», тоді як «00111110011» представляє порожній рядок.

Наведемо реалізацію даного методу в програмі *Mathcad*.

Крок 1

Приймемо, що порожній текстовий контейнер **C** і приховуване у ньому повідомлення **M** є аналогічними використаним у вищерозглянутих методах.

Крок 2

Для уникнення можливих неоднозначностей, зі складу контейнера-оригіналу слід прибрати пробіли, що дублюються. Це можна зробити, наприклад, за допомогою користувальницької функції **dblDel(A, nmbr)**, реалізованої програмним модулем (М.170).

```
dblDel(A, nmbr) := | j ← 1                                          (M.170)
                   | for i ∈ 1.. rows(A)
                   |   | newA_j ← A_i , j ← j+1  if  A_i ≠ nmbr
                   |   | if  1  on error  A_i ≠ A_(i+1)  otherwise
                   |   |   | newA_j ← A_i
                   |   |   | j ← j+1
                   | newA
```

У нашому випадку в якості першого аргументу функції має бути використаний вектор десяткових *ASCII*-кодів символів контейнера ($\mathbf{M}_{vec}$), а в якості другого — число 32 (код пробілу).

Вбудовування повідомлення до контейнера здійснюється програмним модулем (М.171). При цьому одержання вектора двійкового представлення приховуваного повідомлення ($\mathbf{M}_{vec\_bin}$) та підрахунок кількості символів у **r**-му текстовому рядку можна виконати в той самий спосіб, що й у програмному модулі (М.168) (позначено символом «#»).

Або ж можна скористатися перевагами, що їх надає вбудована функція *Mathcad* — **match**(...).

У процесі приховування **μ**-го біта повідомлення $\mathbf{M}_{\mathbf{vec\_bin}}$ обчислюються індекси **i1** та **i2**, що обмежують символи **r**-го текстового рядка в загальному векторі *ASCII*-кодів символів **C**. До рядка (або ж рядків), що має максимальну кількість символів у своєму складі (назвемо його *довгим*) — $\mathbf{L_r} = \mathbf{max}(\mathbf{L})$, — елементи повідомлення не вбудовуються. До всіх інших рядків (назвемо їх *короткими*) — $\mathbf{max}(\mathbf{L}) - \mathbf{L_r} > 0$, — вбудовування бітів виконується таким чином, щоб додавання пробілів розширило поточний рядок до розмірів довгого рядка.

Для виокремленого зі спільного вектора **C** підмасиву **c**, що містить коди символів короткого **r**-го рядка формується масив **space**, розмірність якого відповідає загальній кількості пробілів **t** у цьому рядку, а кожен елемент являє собою порядковий номер відповідних їм пробілів серед усіх символів рядка. Наприклад, для контейнера, зображеного на рис. 5.147, при $\mathbf{r} = 8$ отримуємо $\mathbf{t} = 10$ і

$$\mathbf{space}^{\mathbf{T}} = (7 \quad 17 \quad 24 \quad 28 \quad 35 \quad 39 \quad 41 \quad 52 \quad 57 \quad 61).$$

З опису методу відомо, що один біт даних вбудовується до текстового контейнера шляхом модифікації пари пробілів, які охоплюють собою одне слово речення, — додаванням одного пробілу на початку або в кінці слова, в залежності від значення приховуваного біта. Отже, кількість бітів **Nb**, яку можна вбудувати до **r**-го рядка, визначається співвідношенням між кількістю можливих пар пробілів і кількістю символів, на яку короткий рядок «відстає» від довгого.

Якщо $\mathbf{t}/2 < [\mathbf{max}(\mathbf{L}) - \mathbf{L_r}]$, для вбудовування бітів повідомлення можна використати лише $\mathbf{Nb} := \mathbf{floor}(\mathbf{t}/2) - 1$ пар пробілів; решта ж використовуватимуться для подовження рядка до розмірів довгого — шляхом створення порожніх пар (для спрощення алгоритму, приймається, що видобування на певній ділянці рядка порожньої пари пробілів вказує на те, що у подальшій частині рядка вбудованих даних *не міститься*).

Якщо $\mathbf{t}/2 \geq [\mathbf{max}(\mathbf{L}) - \mathbf{L_r}]$, для вбудовування бітів повідомлення використовується кількість пар пробілів, що дорівнює $[\mathbf{max}(\mathbf{L}) - \mathbf{L_r}]$, адже саме цієї кількості символів не вистачає короткому рядку, щоб зрівнятися з довгим.

Так, приміром, для обраного нами контейнера довгими є 1, 5, 9 і 10-й рядки, що містять по 68 друкованих символів. Решта рядків тексту є короткими. Зокрема, таким є й 8-й рядок: $\mathbf{L}_8 = 62$. Для нього загальна кількість можливих пар пробілів: $\mathbf{t}/2 = 10/2 = 5$, що є меншим за $\mathbf{max}(\mathbf{L}) - \mathbf{L}_8 = 68 - 62 = 6$. Таким чином, $\mathbf{Nb} = 4$.

```
S := | M_vec ← str2vec(M),  dblDel(M_vec, 32)
     | for j ∈ 1.. strlen(M)
     |   M_vec_bin ← if(j = 1, D2B(M_vec_1, 8), stack(M_vec_bin, D2B(M_vec_j, 8)))
     | LF ← match(10, C)
     | for r ∈ 1.. rows(LF)
     |   L_r ← LF_r − if(r = 1, 0, LF_(r−1)) − 2
     | Σ ← 0,  r ← 1,  μ ← 1,  L_max ← max(L),  N_str ← rows(L)
     | while μ ≤ 8·strlen(M)
     |   break if r > N_str
     |   i1 ← if(r = 1, 1, LF_(r−1) + 1),  i2 ← LF_r
     |   c ← subvec(C, i1, i2),  Σ ← Σ + rows(c)
     |   if L_max − L_r > 0
     |     space ← match(32, c),  t ← rows(space)
     |     Nb ← if(t/2 < L_max − L_r, floor(t/2) − 1, L_max − L_r)
     |     τ ← 1
     |     while τ ≤ Nb
     |       break if μ > 8·strlen(M)
     |       sp2_τ ← space_(2·τ)
     |       seg_τ ← subvec(c, if(τ = 1, 0, sp2_(τ−1)) + 1, sp2_τ)
     |       ssp ← match(32, seg_τ)
     |       seg_new ← ins_vec2vec(seg_τ, 32, ssp_(2−M_vec_bin_μ) + 1)
     |       s_r ← if(τ = 1, seg_new, stack(s_r, seg_new))
     |       if τ = Nb
     |         seg_end ← subvec(c, sp2_τ + 1, rows(c))
     |         if t/2 < L_max − L_r
     |           ssp ← match(32, seg_end)
     |           seg_end_new ← ins_vec2vec(seg_end, 32, ssp_1 + 1)
     |           for k ∈ 2.. rows(ssp)
     |             seg_end_new ← ins_vec2vec(seg_end_new, 32, ssp_k + k)
     | ⋮  ⋮  ⋮  ⋮
```

(M.171)

продовження модуля (M.171) — на наступній сторінці

```
: : : :
| | | | seg_end_new ← seg_end  if  t/2 ≥ L_max − L_r
| | | | s_r ← stack(s_r , seg_end_new)
| | | | while  rows(s_r) − 2 < L_max                    if  r < N_str
| | | |   ssp ← match(32 , s_r) ,  k ← rows(ssp)
| | | |   s_r ← ins_vec2vec(s_r , 32 , ssp_k + 1)
| | | μ ← μ + 1 ,  τ ← τ + 1
| | | s_r ← c  if  L_r = L_max
| | | r ← r + 1
| | for  r ∈ 1 .. rows(s)
| |   S ← if(r = 1 , s_1 , stack(S , s_r))
| | S ← stack(S , subvec(C , Σ + 1 , rows(C)))  if  rows(s) < N_str
| | S
```

Після визначення кількості **Nb** пар пробілів, до яких можуть бути вбудовані біти повідомлення, з рядка (підмасиву **c**) виокремлюються сегменти $\mathbf{seg}_\tau$, що, відповідно, містять τ-і пари придатних для вбудовування пробілів. У нашому прикладі, для рядка № 8 одержуються наступні чотири (**Nb** = 4) сегменти (для наочності, зображено відповідні *ASCII*-кодам символи):

$\mathbf{seg}_1 \rightarrow$ [с][о][н][ц][е][м][ ][т][в][р][е][м][н][о][м][у][ ];

$\mathbf{seg}_2 \rightarrow$ [д][в][о][р][і][.][ ][А][л][е][ ];

$\mathbf{seg}_3 \rightarrow$ [б][а][г][а][т][о][ ][х][т][о][ ].

$\mathbf{seg}_4 \rightarrow$ [з][ ][а][р][е][ш][т][а][н][т][і][в].

Для кожного сегмента $\mathbf{seg}_\tau$ формується вектор **ssp**, кожен з елементів якого містить індекси відповідних сегменту пробілів (першого і другого). Наприклад, для зазначених вище чотирьох сегментів:

$$\mathbf{ssp} = \begin{pmatrix} 7 \\ 17 \end{pmatrix} \text{ для } \mathbf{seg}_1; \quad \mathbf{ssp} = \begin{pmatrix} 7 \\ 11 \end{pmatrix} \text{ для } \mathbf{seg}_2;$$

$$\mathbf{ssp} = \begin{pmatrix} 7 \\ 11 \end{pmatrix} \text{ для } \mathbf{seg}_3; \quad \mathbf{ssp} = \begin{pmatrix} 2 \\ 13 \end{pmatrix} \text{ для } \mathbf{seg}_4.$$

За допомогою додатково визначеної функції **ins_vec2vec**(...) формуються нові сегменти, доповнені одним додатковим пробілом або після першого, або після другого пробілу — в залежності від значення поточного вбудовуваного біта $\mathbf{M}_{\mathbf{vec\_bin}_\mu}$.

У загальному випадку функція **ins_vec2vec(A, B, ρ)** (програмний модуль (М.172)) дозволяє проводити вбудовування вектора (або окремого числа) **B** до вектора **A**, починаючи з позиції **ρ**.

$$\begin{aligned}
&\mathbf{ins_vec2vec}(A, B, \rho) := \left|\begin{array}{l}
\mathbf{A1} \leftarrow \mathbf{subvec}(A, 1, \rho - 1) \\
\mathbf{if}\ \ \rho \le \mathbf{rows}(A) \\
\quad \left|\begin{array}{l}
\mathbf{A2} \leftarrow \mathbf{subvec}(A, \rho, \mathbf{rows}(A)) \\
\mathbf{Anew} \leftarrow \mathbf{stack}(\mathbf{A1}, B, \mathbf{A2})
\end{array}\right. \\
\mathbf{Anew} \leftarrow \mathbf{stack}(\mathbf{A1}, B)\ \ \mathbf{if}\ \ \rho = \mathbf{rows}(A) + 1 \\
\mathbf{Anew}
\end{array}\right. \qquad \text{(М.172)}
\end{aligned}$$

У нашому застосуванні, в якості вектора, до якого відбувається вбудовування, виступає сегмент $\mathbf{seg}_\tau$. Вбудовується ж *ASCII*-код пробілу (**B** = 32) після першого ($\rho = \mathbf{ssp}_1 + 1$), якщо $\mathbf{M}_{\mathbf{vec\_bin}_\mu} = 1$, або після другого ($\rho = \mathbf{ssp}_2 + 1$), якщо $\mathbf{M}_{\mathbf{vec\_bin}_\mu} = 0$, пробілу сегмента.

Зокрема, на основі наведених на попередній сторінці сегментів одержуються наступні, модифіковані сегменти[xxx]:

$\mathbf{seg}_{\mathbf{new}_1}$ → сонцем тяремному ;

$\mathbf{seg}_{\mathbf{new}_2}$ → двері. Але ;

$\mathbf{seg}_{\mathbf{new}_3}$ → багато хто .

$\mathbf{seg}_{\mathbf{new}_4}$ → з арештантів .

Одержані нові сегменти формують **r**-й рядок контейнера-результату ($\mathbf{s}_r$).

Після вбудовування останнього (**Nb**-го) дозволеного у даному рядку біта і формування початку рядка $\mathbf{s}_r$, залишається сегмент оригінального рядка **c**, який не зазнав модифікації — $\mathbf{seg}_{\mathbf{end}}$. Для рядка № 8:

$\mathbf{seg}_{\mathbf{end}}$ → саме так і.

Для вирівнювання восьмого рядка по ширині, щоб він зрівнявся за довжиною з найдовшими рядками обраного нами тексту, наразі не вистачає $\mathbf{max}(L) - L_8 - \mathbf{Nb} = 68 - 62 - 4 = 2$ пробіли.

[xxx] Нагадаємо, що саме вбудовування проводиться, починаючи з наймолодших бітів.

Якщо $\mathbf{t}/2 < \mathbf{max}(\mathbf{L}) - \mathbf{L_r}$, то до охоплюючих пробілів завершального сегмента $\mathbf{seg_{end}}$ додаються ще по пробілу, що дозволить після приєднання цього сегмента вже наявної частини рядка $\mathbf{s_r}$ одержати рядок, кількість символів у якому дорівнюватиме $\mathbf{max}(\mathbf{L})$. Якщо ж виконується умова $\mathbf{t}/2 \geq \mathbf{max}(\mathbf{L}) - \mathbf{L_r}$, завершальний сегмент залишається без змін.

Для розгляданого вище прикладу, оскільки $\mathbf{t}/2 = 5 < \mathbf{max}(\mathbf{L}) - \mathbf{L_r} = 6$, останній сегмент зазнає модифікації додаванням пробілів, після якої набуває наступного вигляду:

$$\mathbf{seg_{end\_new}} \rightarrow \boxed{\text{саме}\ \ \text{так}\ \ \text{і}}.$$

Як вже зазначалося, не залежно від того, був чи не був модифікованим завершальний сегмент, він дописується в кінець рядка $\mathbf{s_r}$. Одержані рядки об'єднуються до спільного масиву контейнера-результата **S**.

Паралельно, у програмному модулі (М.171) проводився підрахунок кількості символів оброблених рядків $\mathbf{\Sigma}$. Це дозволяє визначити індекс символу оригінального тексту, після якого останній не зазнав змін (через завершення вбудовуваного повідомлення раніше, аніж було досягнуто кінець тексту), і дописати немодифікований текст в кінець контейнера.

Фрагмент результату вбудовування повідомлення **M** до текстового контейнера **C** наведено на рис. 5.150. При цьому конструкція виду «пробіл-пробіл-слово-пробіл» є еквівалентом двійкової «1», конструкція «пробіл-слово-пробіл-пробіл» — еквівалент «0», а «пробіл-пробіл-слово-пробіл-пробіл-...» — уособлює собою пустий символ.

Широкий прохід по центру блока «Д» був застелений лінолеумом кольору попсутих лаймів. З цієї причини те, що в інших в'язницях зазвичай називали Останньою Милею, у «Холодній Горі» було відомим як Зелена Миля. Остання тягнулася приблизно на шістдесят широких кроків з півдня на північ. На її початку знаходився ізолятор, завершувалася ж вона Т-подібним перехрестям. Поворот ліворуч на цьому «роздоріжжі» означав життя – якщо можна назвати життям прогулянки на висушеному сонцем тюремному дворі. Але багато хто з арештантів саме так і вважав: вони ж бо роками жили в таких умовах без видимих згубних для себе наслідків. Жиґани, підпалювачі, гвалтівники – и усі ведуть свої розмови, ходять на свої прогулянки, проветрюють свої маленькі оборудки. А ось поворот праворуч – то вже, як то кажуть, «відчуйте різницю». Спочатку ви потрапляєте до мого робочого кабінету (кислим на долівці в якому, до речі, також зеленого кольору – та річ, яку я все збирався змінити, але так і не викроїв на це часу) і крокуєте ним повз мій стіл, за яким стояло два прапори: ліворуч – державний, праворуч – штату. У дальньому кінці кабінету було двоє дверей: за однією ховалася невеличка вбиральня, якою користався я та інші охоронці з блока (а час від часу – навіть наш шеф Мурс), за іншою – щось типу складського приміщення. Саме останнє було б вашою кінцевою зупинкою у прогулянці Зеленою Милею.

Рис. 5.150. Фрагмент текстового контейнера-результату, заповненого методом зміни типу пробілів в кінці рядків

Неважко підрахувати, що в зображеному на рис. 5.147 фрагменті тексту розміром **rows**(**C**) = 1394 байти (11152 біти) приховано 38 бітів повідомлення (це майже у 4 рази більше, аніж було приховано методом з підп. 5.5.1.1, і вдвічі менше за метод з підп. 5.5.1.2), що відповідає трьом повним байтам або першим трьом символам повідомлення **M**: "© Pu". Так, у рядах №№ 1, 5, 9 і 10 не приховано жодного біта і не додано жодного пробілу, бо вони первинно були довгими. У рядку № 2 приховано біти «100»; у рядку № 3 приховано біти «10»; у рядку № 4 приховано біти «101» (отож, маємо символ "©"). У рядку № 6 приховано біти «00»; у рядку № 7 приховано біти «00»; у рядку № 8 приховано біти «0100» (маємо символ пробілу " "). У рядку № 11 приховано біти «00»; у рядку № 12 приховано біти «00»; у рядку № 13 приховано біти «10»; у рядку № 14 приховано біт «1»; у рядку № 15 приховано біти «0» (маємо символ "P") і «10»; у рядку № 16 приховано біти «1»; у рядку № 17 приховано біти «011»; у рядку № 17 приховано біти «10» (маємо символ "u") і «0» — НЗБ наступної літери повідомлення, — і т. д.

Крок 3

Видобування з контейнера $\mathbf{S}^* := \mathbf{S}$ стеганографічно вбудованих до нього бітів повідомлення здійснюється за допомогою програмного модуля (М.173).

На початку модуля виконується підрахунок кількості символів у кожному з рядків тексту $\mathbf{S}^*$. На підставі одержаного при цьому масиву **L** обчислюються індекси **i1** та **i2**, що обмежують **r**-й текстовий рядок у спільному векторі *ASCII*-кодів $\mathbf{S}^*$.

Для виокремленого з вектора $\mathbf{S}^*$ рядка-підмасиву **s** формується вектор **space**, розмірність якого відповідає загальній кількості пробілів у даному рядку, а елементи містять інформацію про порядкові номери відповідних їм пробілів серед множини символів рядка.

На основі одержаного вектора **space** формується вектор **b**, значення кожного елемента якого залежить від результату обчислення різниці між наступним і поточним значеннями елементів вектора **space**. Якщо ця різниця дорівнює одиниці, елемент $\mathbf{b}_j$ приймає значення 1. Якщо ж різниця є більшою за одиницю, елементу $\mathbf{b}_j$ присвоюється значення 0.

З одержаного вектора **b** виокремлюються підмасиви з трьох елементів (так звані трибіти), на основі яких робиться висновок про поточне значення вбудованого біта: якщо трибіт являє собою послідовність (0 1 0) — було приховано біт «0»; якщо ж (1 0 0) — біт «1». В усіх інших випадках приймається рішення про відсутність вбудованих бітів. Сформована при цьому двійкова послідовність перетворюється на рядок символьних даних, який і повертається змінній $\mathbf{M}^*$.

```
M* := | LF ← match(10, S*)
      | for r ∈ 1..rows(LF)
      |   L_r ← LF_r − if(r = 1, 0, LF_{r−1}) − 2
      | r ← 1, μ ← 1, N_str ← rows(L)
      | while r ≤ N_str
      |   i1 ← if(r = 1, 1, LF_{r−1} + 1), i2 ← LF_r
      |   s ← subvec(S*, i1, i2)
      |   space ← match(32, s)
      |   t ← rows(space)
      |   for j ∈ 1..t − 1
      |     b_j ← 0 if (space_{j+1} − space_j) > 1
      |     b_j ← 1 if (space_{j+1} − space_j) = 1
      |   for τ ∈ 1.. rows(b)/3
      |     tribit ← subvec(b, 3·τ − 2, 3·τ)
      |     M*bin_μ ← 0, μ ← μ + 1 if tribit^T = (0 1 0)
      |     M*bin_μ ← 1, μ ← μ + 1 if tribit^T = (1 0 0)
      |     break otherwise
      |   r ← r + 1
      | for j ∈ 1.. rows(M*bin)/8
      |   M*vec_j ← B2D(subvec(M*bin, 8·j − 7, 8·j))
      |   M*vec_j ← 32 if M*vec_j < 32 ∧ M*vec_j ≠ 10 ∧ M*vec_j ≠ 13
      | vec2str(M*vec)
```

(M.173)

Розглянуті реалізації методів довільного інтервалу є ефективними, за умови, що первинний текст було представлено у кодах *ASCII*. Як вже зазначалося вище, після роздрукування тексту деякі вбудовані дані можуть виявитися втраченими. Друковані документи висувають до приховування даних такі вимоги, що далеко виходять за можливості текстового файлу при *ASCII*-кодуванні. При цьому приховування даних у «жорстких» копіях тексту може виконуватися шляхом незначних змін відстані між словами й окремими літерами, зміною позицій базових ліній (ліній, на яких лежать найнижчі елементи літер або знаків пунктуації рядка), зміною форм літер тощо.

5.5.2. Синтаксичні й семантичні методи

Той факт, що вільне місце для вбудовування у текст обирається довільно, є одночасно як перевагою, так і недоліком з точки зору стійкості прихованих даних. Пересічний читач може навіть і не помітити проведені з текстом маніпуляції, тоді як текстовий редактор здатен автоматично змінити кількість і розміщення пробілів, руйнуючи таким чином стеганограму.

Низька стійкість до атак, в світлі можливого переформатування документа, є однією з причин пошуку для вбудовування повідомлень до текстових контейнерів інших методів. Яскравим прикладом останніх, зокрема, є синтаксичні й семантичні методи, які, крім того, що взагалі жодним чином не використовують вільні місця в тексті, докорінно відрізняються від вищерозглянутих. Разом з тим, усі ці методи можуть використовуватися одночасно, дублюючи або доповнюючи один одного.

До *синтаксичних методів* текстової стеганографії відносяться методи зміни пунктуації та методи зміни структури й стилю тексту [14]. Існує чимало випадків, коли правила пунктуації є неоднозначними, і відхід від них на загальний зміст тексту впливає несуттєво. Так, приміром, фрази *«червоний, зелений, синій»* та *«червоний, зелений і синій»* є еквівалентними одна одній. Той факт, що вибір подібних форм може бути цілком довільним[XXXI], й використовується при побудові стеганосистем на основі синтаксичних методів, оскільки періодична зміна форм може бути поставлена у відповідність до двійкових даних. Наприклад, поява в тексті форми перерахування із сполучником «*і*» розуміє під собою вбудований біт «1», у той час як відсутність сполучника при перерахуванні говорить про те, що було вбудовано біт «0». Серед інших прикладів можна навести використання скорочень і абревіатур. Середня швидкість передавання даних такими методами не перевищує декількох бітів на один кілобайт тексту [14].

Разом з тим, у той час як писемна мова надає достатньо можливостей для синтаксичного приховування даних, такі можливості суттєво обмежені у відомих прозаїчних творах. Крім того, хоча деякі з правил пунктуації і є неоднозначними, їхнє суперечливе використання може стати об'єктом уваги для цензора. Також можливі й випадки, коли зміна пунктуації призводить до зниження зрозумілості тексту або ж до набування ним діаметрально протилежного змісту. Тому авторами [14] рекомендується застосовувати цей метод з обачністю.

[XXXI] Але, звичайно, довільним з огляду на використовуваний в якості контейнера текст, оскільки зрозуміло, що стеганографічна система, побудована на основі видозміни широковідомого тексту (класичного твору або ще) навряд чи може вважатися надійною

До синтаксичних методів також відносяться методи зміни стилю й структури тексту без значної зміни його смислового навантаження. Наприклад, речення *«Існує чимало випадків, коли правила пунктуації є неоднозначними»* можна сформулювати як *«Правила пунктуації є неоднозначними у багатьох випадках»*. Такі методи, порівняно з методами зміни пунктуації, є більш непомітними для непосвячених, але можливість їх використання обмежена складністю автоматизування процесу стеганографічного вбудовування/видобування бітів повідомлення.

Семантичні методи є подібними до синтаксичних. Замість того, щоб вбудовувати двійкові дані, використовуючи двозначність граматичної форми, такі методи призначають два синоніми, що відповідають значенням приховуваних бітів. Наприклад, слово *«проте»* може бути поставлене у відповідність до «1», а слово *«однак»* — до «0».

Для проведення приховування з використанням семантичних методів необхідна наявність узгодженої таблиці синонімів. Крім того, як зазначено у [14], якщо слову відповідає досить велика кількість синонімів, виникає можливість одночасного кодування більшої кількості бітів. Скажімо, вибір між синонімами *«секретний»*, *«таємний»*, *«прихований»*, *«конфіденційний»*, *«негласний»*, *«замаскований»*, *«невідомий»*, *«приватний»* дає можливість за одне вбудовування приховати трійку бітів даних. Проблеми можуть виникнути, однак, коли бажанню вбудувати фрагмент повідомлення перешкоджатиме певний нюанс значення слова.

5.6. Системні вимоги

Для ефективного застосування запропонованих у цьому розділі програмних модулів реалізації методів комп'ютерної стеганографії рекомендується робоче місце, системні вимоги до якого зумовлені вимогами, що їх висуває система комп'ютерної алгебри *Mathcad*[XXXII]: операційна система *Microsoft Windows 8*, *Windows 7* (64- або 32-розрядна), *Windows Vista* (64- або 32-розрядна), *Windows XP* (у тому числі й 64- розрядна) і Windows 2000. Процесор або система на чипі з тактовою частотою 700 МГц і вище (рекомендується 2000+ МГц). Оперативна пам'ять 512 МБ; 1.75 ГБ вільного місця на жорсткому диску; встановлений *Microsoft .NET Framework 4.0* або новіше.

Для формування/обробки зображень використовується будь-який графічний редактор. У більшості випадків достатнім є наявність графічного редактора *MS Paint*, що поставляється разом з ОС *MS Windows* і дозволяє працювати з точковими зображеннями форматів *JPG*, *GIF*

[XXXII] Див., наприклад, http://mathcad.com.ua/down-math.php

або *BMP*. За допомогою *MS Paint* можна моделювати такі спотворення як масштабування зображення, його поворот, нахилення, фрагментація, відображення по горизонталі чи вертикалі, інвертування кольорів. Для можливості моделювання атак на зразок додавання гаусівського шуму, фільтрації, експозиції, зміни глибини кольорів, масштабування зображення з використанням різних типів фільтрів передискретизації тощо крім власне вбудованих функцій *Mathcad* по обробці зображення, рекомендується наявність одного зі спеціалізованих пакетів обробки зображення: *ACD FotoCanvas*, *Adobe Photoshop*, *Deformer*, *Focus Photoeditor*, *GIMP*, *JPEG Imager*, *PhotoStudio* або ін.

Для роботи з аудіостеганометодами в більшості випадків достатньо можливостей програми *Звукозапис/Sound (Voice) Recorder*, що є прикладною програмою з інструментарію ОС *MS Windows*. Дана програма дозволяє здійснювати запис, змішування, відтворення й редагування звукозаписів. Крім того, вона дозволяє виконувати наступні операції над імпортованим аудіофайлом: видалення частини звукозапису, зміна швидкості і гучності відтворення, зміна напрямку відтворення, зміна і конвертація типів звукозаписів, додавання луни. Для моделювання спотворень рекомендується використовувати такі аудіоредактори як *WaveShop*, *Nero Wave Editor*, *AudioEdit Deluxe*, *SoundForge*, *GoldWave*, *AV Voice Changer Software*, *Audio Editor Gold* або ж подібні.

Для дослідження методів лінгвістичної стеганографії достатньо таких текстових редакторів як *WordPad*, *TextMaker*, *MS Office Word* тощо.

5.7. Підсумки розділу

У даному розділі було докладно розглянуто типові стеганографічні методи приховання даних у нерухомих зображеннях, в аудіосигналах і текстових файлах. Наведено приклади програмних комплексів, що демонструють принципи, закладені до основи більш як двадцяти методів стеганографічного приховування інформації в просторовій, часовій, частотній областях використовуваних мультимедійних контейнерів. Розробку комплексів було проведено з використанням популярної математичної системи *Mathcad*.

Усі етапи приховування супроводжені відповідними програмними модулями (більше 170). Отримані авторами під час моделювання результати належним чином проілюстровані. Наведено наслідки можливих атак на стеганоповідомлення та рекомендації щодо захисту від них. Також обчислено основні показники візуального та звукового спотворень, що дозволяє проводити аналіз оптимальності обраного формату контейнера для приховування визначених типів даних.

Розділ 6. СТЕГАНОАНАЛІЗ ЦИФРОВИХ ЗОБРАЖЕНЬ

6.1. Виявлення стеганографічних модифікацій процедур формування цифрових зображень

У перших відомих алгоритмах приховання повідомлень у цифрових зображеннях (ЦЗ) формування стеганограм проводилося шляхом заміщення службових атрибутів зображень-контейнерів на вбудовувану інформацію [143]. Для виявлення таких стеганограм проводиться аналіз ЦЗ з використанням бази даних відомих демаскуючих ознак (сигнатур) стеганографічних методів (СМ) [40, 144].

Для підвищення стійкості стеганограм до пасивного стеганоаналізу (ПСА) були запропоновані модифіковані методи обробки цифрових зображень, засновані, зокрема, на квантуванні яскравості пікселів графічного контейнера — наприклад, методи стохастичної модуляції, модуляції індексу квантування, використання дизерингуючих[XXXIII] квантувачів [143, 145]. Виявлення стеганограм, сформованих згідно зазначених СМ, проводиться з використанням ефективних методів ПСА, заснованих на аналізі гістограм [146, 147] та характеристичних функцій гістограм [148] розподілу значень яскравості пікселів ЦЗ, кореляції значень яскравості суміжних пікселів ЦЗ на основі теорії марківських процесів [149].

Поява значної кількості ефективних методів виявлення стеганографічних даних (СД), вбудованих шляхом модифікації стандартних етапів обробки ЦЗ-контейнерів (ЦЗК) [143], суттєво обмежила практичне застосування таких СМ. Для підвищення стійкості стеганограм було запропоновано приховувати повідомлення безпосередньо в просторовій області ЦЗК [5, 143] шляхом модифікації НЗБ значень яскравості пікселів ЦЗ (т. зв. *LSB*-методи).

6.2. Виявлення стеганограм, прихованих у просторовій області цифрових зображень

Перші методи приховання повідомлень в просторовій області графічного контейнера були засновані на заміні значень НЗБ яскравості пікселів ЦЗК $\boldsymbol{I}_{x,y}$, розмірами $M \times N$ пікселів, на біти СД [143]:

[XXXIII] Дизерінг (англ. *dither* – тремтіння) — в обробці цифрових сигналів являє собою підмішування до первісного сигналу псевдовипадкового шуму зі спеціально підібраним спектром, метою чого є розупорядкування шуму квантування, запобігання появі небажаних спотворень, гірших для сприйняття. Дизерінг застосовується при обробці графічної, відео- та аудіоінформації.

$$S_{x_i y_i}[n] = d_i,\ \ i \in [1; L_M],\ \ n \in [1; C_l], \tag{6.1}$$

$$S_{x_i y_i} = \sum_{k=1}^{C_l} S_{x_i y_i}[k] \times 2^{C_l - k},$$

де d_i — i-й біт приховуваного повідомлення D довжиною L_D бітів; C_l — розрядність бітового представлення яскравості пікселів ЦЗ; $S_{x_i y_i}[n]$ — значення n-го біта двійкового представлення яскравості i-го пікселя ЦЗ, використовуваного при вбудовуванні елемента повідомлення; $S_{x_i y_i}$ — десяткове представлення яскравості i-го пікселя контейнера-результату.

Вестфельдом (*Westfeld*) та Пфіцманом (*Pfitzman*) було запропоновано перший ефективний метод виявлення факту використання *LSB*-методів — *Pairs-of-Value* (*PoV*) аналіз [144, 145]. Метод заснований на аналізі виду гістограм розподілу значень яскравості пікселів зображень, що містили СД. Суттєвим обмеженням практичного застосування *PoV*-аналізу є те, що виявлення стеганограм з високою ймовірністю можливе лише у випадку суттєвого заповнення контейнера (більше 50%). Для підвищення ж імовірності виявлення стеганограм у випадку слабкого заповнення (менше 10%) було запропоновано використовувати χ^2-тест [150, 151], *RS(Regular-and-Singular)*-аналіз [152] та аналіз зміни центра мас характеристичної функції гістограми (*histogram characteristic function center of mass*) [148].

Для забезпечення високої стійкості стеганограм до методів ПСА на основі χ^2-тесту та *RS*-аналізу був розроблений удосконалений *LSB*-метод (±1) вбудовування СД до цифрових зображень [143]. Відмінністю методу (±1) від класичного *LSB*-методу (6.1) є те, що для приховання повідомлень змінюють значення яскравості лише тих пікселів ЦЗК, НЗБ яких відрізняються від значення вбудовуваного біта.

Для зменшення об'єму бітового представлення СД і, відповідно, кількості змінених пікселів графічного контейнера при формуванні стеганограм згідно методу (±1), були запропоновані матричні *LSB*-алгоритми, засновані на використанні поширених методів кодування — наприклад, алгоритму Гаффмана, гратчастих (або т. зв. треліс-) кодів, *LT(Luby Transform)*-кодів [143, 153]. Подальше зниження демаскуючих ознак стеганограм, сформованих згідно матричних *LSB*-алгоритмів, — мінімізація змін статистичних характеристик графічного контейнера при вбудовуванні окремих бітів — було досягнуто за рахунок використання методів статистичної фізики, наприклад, алгоритму формування псевдовипадкових вибірок (алгоритму Гіббза) [154]. Це дало можливість суттєво збільшити стійкість отримуваних стеганограм до відомих методів ПСА.

Для підвищення ймовірності виявлення стеганограм, сформованих з використанням матричних *LSB*-методів, в [155] був запропонований

SP(Sample-Pairs)-аналіз послідовностей (кортежів) НЗБ кодів яскравості пікселів графічних контейнерів з використанням теорії марківських процесів. Проте, можливість дослідження лише шумових компонентів ЦЗК (якими, по суті, є НЗБ яскравості його пікселів) обмежує практичне застосування *SP*-аналізу у випадку використання сучасних адаптивних *LSB*-методів, наприклад, алгоритмів *HUGO (Highly Undetectable steGO)* [156, 157], *S-UNIWARD (Spatial UNIversal WAvelet Relative Distortion)* [158]. Для подолання зазначеного обмеження *SP*-аналізу Т. Певні [159] було запропоновано використовувати статистичну модель контейнера (СМК) *SPAM (Subtractive Pixel Adjacency Matrix)* зображень у просторовій області. ПСА ЦЗ з використанням *SPAM* проводиться на основі оцінок кореляції між різницями яскравості суміжних пікселів, визначених із застосуванням марківських ланцюгів (МЛ) 1-го та 2-го порядків. Для оцінки параметрів МЛ для півтонового зображення $\boldsymbol{I}_{x,y}$, розмірами $M \times N$ пікселів використовуються матриці суміжності (*co-occurrence matrix*) різниць значень яскравості суміжних пікселів $\boldsymbol{C}_{\Delta x,\Delta y}$ [160]:

$$\boldsymbol{C}_{\Delta x,\Delta y}(i,j) = \sum_{m=1}^{M}\sum_{n=1}^{N}\Big(\big[\boldsymbol{I}_{m,n} - \boldsymbol{I}_{m+\Delta x,n+\Delta y} = i\big]_l \times \big[\boldsymbol{I}_{m+\Delta x,n+\Delta y} - \boldsymbol{I}_{m+2\Delta x,n+2\Delta y} = j\big]_l\Big),$$

де $[a]_l = \begin{cases} 1, a = true \\ 0, a = false \end{cases}$ — нотація (дужка) Айверсона; $(i,j) \in \{[1; Z_l] \times [1; Z_l]\}$ — поточна позиція елемента матриці суміжності $\boldsymbol{C}_{\Delta x,\Delta y}$; $\Delta x, \Delta y$ — просторовий зсув між пікселями ЦЗ; $Z_l = (2^k - 1)$ — діапазон значень яскравості пікселів ЦЗ; k — глибина кольору ЦЗК $\boldsymbol{I}_{x,y}$ у бітах.

Розрахунок матриці $\boldsymbol{M}_{\Delta x,\Delta y}$ ймовірностей переходу між елементами МЛ (різниць значень яскравості суміжних пікселів) проводиться шляхом нормалізації отриманої матриці суміжності $\boldsymbol{C}_{\Delta x,\Delta y}$:

$$\boldsymbol{M}_{\Delta x,\Delta y}(i,j) = \frac{\boldsymbol{C}_{\Delta x,\Delta y}(i,j)}{\sum_{i=1}^{Z_l}\sum_{j=1}^{Z_l}\boldsymbol{C}_{\Delta x,\Delta y}(i,j)}. \qquad (6.2)$$

Для усереднення параметрів моделі *SPAM* використовуються матриці $\boldsymbol{M}_{\Delta x,\Delta y}$ для горизонтального $(\Delta x \neq 0, \Delta y = 0)$, вертикального $(\Delta x = 0, \Delta y \neq 0)$ і діагонального $(\Delta x \neq 0, \Delta y \neq 0)$ напрямків, в яких розраховуються значення кореляції різниць яскравості суміжних пікселів. Загальна кількість параметрів моделі *SPAM* є рівною $d_{SPAM} = 686$. Алгоритм визначення параметрів моделі *SPAM* ЦЗ $\boldsymbol{I}_{x,y}$ — значень елементів матриці $\boldsymbol{M}_{\Delta x,\Delta y}$ (6.2) ймовірностей переходу між елементами МЛ — можна знайти серед додатків [161].

Використання моделі *SPAM* дало можливість суттєво підвищити точність виявлення стеганограм, сформованих згідно сучасних методів приховання повідомлень у просторовій області графічного контейнера, та послужило поштовхом до розробки сучасних СМК: моделей *SRM (Spatial-domain Rich Model)* [162], *J+SRM (JPEG- + Spatial-domain Rich Model)* [163], *CRM (Color Rich Model)* [164], а також алгоритмів *UNIWARD* [158], *UED (Uniform Embedding Distortion metric)* [165], *MVGG (Multivariate Generalized Gaussian algorithm)* [166] і *Synch* [167]. Обмеженням практичного застосування відомих СМК є необхідність аналізу значної кількості тестових зображень для визначення елементів простору ознак великої розмірності — наприклад, 12753 параметрів для *SRM-Q1* моделі [162], 35263 ознак для *J+SRM* моделі [163], — що збільшує тривалість процедури налаштування СД та ускладнює адаптацію СМК для виявлення нових стеганографічних методів.

Для підвищення стійкості стеганограм до методів ПСА Е. Кохом (*E. Koch*) [86, 87] було запропоновано проводити вбудовування СД до областей перетворення ЦЗК (ОПЦЗК) шляхом модифікації коефіцієнтів двовимірного ДКП графічного контейнера (див. підп. 5.3.3.1). Вибір останніх у блоках контейнера проводиться на основі вимог щодо стійкості стеганограм до пасивного (використання ВЧ-піддіапазону) й активного (використання НЧ-піддіапазону) стеганоаналізу. Згідно рекомендацій [86, 87], для вбудовування СД обираються позиції тих коефіцієнтів, що відповідають середнім частотам, — це дозволяє досягти компромісу між стійкістю прихованих повідомлень до ПСА та появою візуальних спотворень контейнера.

6.3. Виявлення стеганограм на основі приховання в областях перетворення цифрових зображень

6.3.1. Детектування повідомлень, прихованих у частотній області цифрових зображень

Висока робастність стеганограм, сформованих згідно методу Коха [87], до методів ПСА призвела до бурхливого розвитку алгоритмів приховання у частотній області ЗК. Одними з найбільш відомих методів формування стеганограм із застосуванням двовимірного ДКП ЦЗК є алгоритми *JSteg* [145], *F5* [168], *nsF5 (no-shrinkage F5)* [143], *OutGuess* [169], *YASS* [170] та *MB* [171].

Для детектування стеганограм з даними, вбудованими в частотній області ЦЗК, були запропоновані методи виявлення характерних змін коефіцієнтів блокового двовимірного ДКП ЦЗК, що призводили до появи візуальних спотворень окремих блоків розбиття стеганограм при

проведенні *JPEG*-стиснення (т. зв. *blockiness*-ефект) [172], та аналізу гістограм розподілу значень коефіцієнтів двовимірного ДКП ЦЗ (гістограмна атака) [143]. Обмеженнями даних методів є велика кількість помилок першого роду (хибного віднесення ЦЗ до класу стеганографічно заповненого контейнера), обумовлена особливостями використання *JPEG*-методу стисненням зображень [173].

Для підвищення ймовірності виявлення стеганограм, сформованих згідно сучасних методів приховання повідомлень в частотній області ЦЗ з використанням двовимірного ДКП, наприклад, вже згаданих вище алгоритмів *nsF5* та *YASS*, було запропоновано використовувати статистичні моделі зображень-контейнерів у частотній області [143]. Однією з найбільш відомих СМК у спектральній області є модель *CC-PEV (Cartesian Calibrated model by Tomáš PEVný)* [174]. При проведенні ПСА ЦЗ на основі моделі *CC-PEV* проводиться попередня обробка стеганограм для придушення спотворень характеристик графічного контейнера, зумовлених послідовним *JPEG*-стисненням з використанням методу калібрування контейнера [173, 172]. Обробка ЦЗ згідно даного методу полягає в декомпресії *JPEG*-зображення, видаленні (кадруванні) перших чотирьох рядків і стовпців отриманого зображення та повторного *JPEG*-стиснення кадрованого ЦЗ з використанням заданої таблиці квантування.

Визначення кореляційних характеристик зображення $\boldsymbol{I}_{x,y}$ розмірності $M \times N$ пікселів проводиться у два етапи [174]. На першому етапі ЦЗ $\boldsymbol{I}_{x,y}$ розбивається на блоки розміром $B \times B$ пікселів, що не перетинаються, після чого до кожного блока застосовується двовимірне ДКП [160]. У випадку, якщо розміри зображення $\boldsymbol{I}_{x,y}$ не є кратними до розмірів блока розбиття ($M \bmod B \neq 0$, $N \bmod B \neq 0$), воно доповнюється M_{add} рядками та N_{add} стовпцями з нульовими елементами ($(M + M_{add}) \bmod B = 0$, $(N + N_{add}) \bmod B = 0$). Згідно рекомендацій [121, 160, 174], розміри блоків розбиття зображення $\boldsymbol{I}_{x,y}$ є рівними $B = 8$ пікселів.

На другому етапі проводиться розрахунок параметрів моделі *CC-PEV*, які можуть бути розділеними на дві підгрупи [174]:

1) Статистичні характеристики коефіцієнтів двовимірного ДКП всередині блоків розбиття зображення:

а) локальні гістограми $\boldsymbol{h}^{ij}$ розподілу НЧ-коефіцієнтів двовимірного ДКП — розраховуються для значень перших п'яти НЧ-коефіцієнтів двовимірного ДКП по всім блокам розбиття ЦЗ;

б) матриці $\boldsymbol{C}_{st}$ суміжності значень коефіцієнтів двовимірного ДКП, квантованих у діапазоні $d \in [-2; 2]$ з кроком $\Delta_d = 1$, в блоках розбиття зображення $\boldsymbol{I}_{x,y}$;

в) матриця $\bar{\boldsymbol{M}}$ імовірності переходу між елементами МЛ (коефіцієнтами двовимірного ДКП), квантованих у діапазоні $d \in [-4; 4]$

з кроком $\Delta_d = 1$ — використовується для оцінки величини кореляції значень коефіцієнтів двовимірного ДКП всередині блоків розбиття зображення $\boldsymbol{I}_{x,y}$ з використанням МЛ першого порядку;

2) *Статистичні характеристики коефіцієнтів двовимірного ДКП між блоками розбиття зображення:*

а) глобальна гістограма $\boldsymbol{H}$ розподілу коефіцієнтів двовимірного ДКП — розраховується з використанням коефіцієнтів двовимірного ДКП по всіх блоках розбиття зображення $\boldsymbol{I}_{x,y}$;

б) глобальні гістограми $\boldsymbol{g}_{ij}^{d}, i,j \in [1;8]$ розподілу коефіцієнтів двовимірного ДКП, квантованих у діапазоні $d \in [-5;5]$ з кроком $\Delta_d = 1$, на позиціях $i,j \in \{(1,2), (1,3), (1,4), (2,1), (3,1), (4,1), (2,2), (2,3), (3,2)\}$;

в) усереднена варіація V значень коефіцієнтів двовимірного ДКП у суміжних блоках розбиття зображення;

г) функціонали $B_{\alpha_{DCT}}, \alpha_{DCT} \in \{1,2\}$ — використовуються для оцінки величини *blockiness*-ефекту при *JPEG*-стиснені ЦЗ $\boldsymbol{I}_{x,y}$.

Для визначення вищенаведених характеристик ЦЗ використовуються коефіцієнти двовимірного ДКП досліджуваного зображення, що належать інтервалу $[\mu_{DCT} - 3\sigma_{DCT}^2; \mu_{DCT} + 3\sigma_{DCT}^2]$, де $\mu_{DCT}, \sigma_{DCT}^2$ — відповідно, середнє значення і дисперсія значень коефіцієнтів двовимірного ДКП. Для досягнення компромісу між точністю визначення параметрів розподілу коефіцієнтів двовимірного ДКП досліджуваного зображення та обчислювальною складністю алгоритму обробки ЦЗ інтервал значень коефіцієнтів двовимірного ДКП розділюється на $N_h = 11$ піддіапазонів [173]. Для підвищення ймовірності виявлення стеганограм з СД, вбудованими в частотній області графічного контейнера, локальні й глобальні характеристики значень двовимірного ДКП розраховуються як для вихідного зображення $F(\boldsymbol{I}_{x,y})$, так і для його каліброваної версії — $F(\boldsymbol{I}_{x,y}^{c})$. Унаслідок цього кількість параметрів моделі *CC-PEV* є рівною $d_{CC-PEV} = 2 \times (N_h \times 15 + 109) = 548$ [173]. З алгоритмом визначення параметрів моделі *CC-PEV* зображення $\boldsymbol{I}_{x,y}$ можна ознайомитися, наприклад, у [161].

Стандартним підходом до виявлення стеганограм з СД, вбудованими до частотної області графічного контейнера з використанням двовимірного ДКП, є розрахунок значень параметрів статистичної моделі *CC-PEV* – статистичних характеристик коефіцієнтів двовимірного ДКП всередині та між блоками розбиття ЦЗ — для випадку *JPEG*-стиснення стеганограм без втрат (*İQF* = 100) та зі втратами (*İQF* = 90) [173], де *İmage Quality Factor (İQF)* — коефіцієнт масштабування елементів таблиці квантування коефіцієнтів двовимірного ДКП при проведенні

JPEG-стиснення ЦЗ [160, 175]. Це дозволяє виявляти характерні зміни статистичних характеристик коефіцієнтів двовимірного ДКП ЦЗ, зумовлені вбудовуванням СД.

На основі моделі *CC-PEV* були розроблені сучасні статистичні моделі цифрових зображень у частотній області, зокрема, такі моделі як *CC-C300* [176], *CC-JRM (Cartesian Calibrated JPEG Rich Model)* [163] та *J+SRM* [163].

Одним з найбільш складних випадків ПСА ЦЗ є виявлення СД, вбудованих до ОПЦЗК з використанням як спектральних перетворень (зокрема, двовимірних дискретних косинусних та вейвлет перетворень), так і спеціальних (наприклад, сингулярного розкладу) перетворень матриць яскравості пікселів ЦЗК.

6.3.2. Методи приховання повідомлень до області перетворення цифрових зображень

Одними з найбільш поширених методів приховання повідомлень до ОПЦЗК є одноетапні методи Дея [177] і Агарвала [178], багатоетапні методи Джозефа [179] і Хана [180], а також комплексні методи Елахіана [181] і Ганджела [182]. Висока робастність стеганограм, сформованих згідно одноетапних методів, до активного стеганоаналізу (АСА) — фільтрації, стиснення зображень зі втратами — забезпечується шляхом застосуванням до графічного контейнера та СД двовимірного дискретного вейвлет перетворення та сингулярного розкладу. Використання декількох етапів обробки графічного контейнера у багатоетапних СМК дозволяє досягти компромісу між робастністю отримуваних стеганограм до ПСА та стійкістю прихованих повідомлень до випадкових (вплив шумів каналу зв'язку) або навмисних (застосування методів АСА) модифікацій контейнера. Для комплексних методів приховання повідомлень до ОПЦЗК характерна наявність етапу попередньої обробки СД з метою наближення виду останніх до псевдовипадкового сигналу, що ускладнює проведення ПСА контейнерів-результатів.

Формування стеганограм згідно одноетапних, багатоетапних і комплексних методів проводиться шляхом вагового додавання коефіцієнтів розкладу ЦЗК $W(\boldsymbol{I})$ та СД $W(\boldsymbol{D})$, представлених у вигляді ЦЗ, в обраному базисі перетворення $W(\cdot)$ з ваговим коефіцієнтом G, що залежить від енергії приховуваних повідомлень:

$$W(\boldsymbol{S}) = W(\boldsymbol{I}) + G \times W(\boldsymbol{D}), \tag{6.3}$$

де $W(\boldsymbol{S})$ — матриця коефіцієнтів заповненого графічного контейнера.

Як вже було зазначено, обробка окремих матриць (колірних каналів) зображення-контейнера та стеганографічних даних згідно одноетапних,

багатоетапних та комплексних методів приховання повідомлень до ОПЦЗК проводиться з використанням стандартних та спеціальних перетворень — двовимірних дискретних вейвлет [160, 183] та косинусного [160] перетворень, а також сингулярного розкладу [184].

Для формування стеганограми в просторовій області до отриманих коефіцієнтів $W(\boldsymbol{S})$ (6.3) застосовується зворотне перетворення — $W^{-1}\big(W(\boldsymbol{S})\big)$. Вилучення (екстракція) СД проводиться згідно виразу:

$$W(\boldsymbol{D}) = \big(W(\boldsymbol{S}) - W(\boldsymbol{I})\big)/G.$$

Формування стеганограм $\boldsymbol{S}_{M\times N}$ згідно одноетапного методу Агарвала проводиться з використанням сингулярного розкладу матриць яскравості пікселів ЦЗК $\boldsymbol{I}_{M\times N}$ та СД $\boldsymbol{D}_{M\times N}$, представлених у вигляді півтонових ЦЗ розміром $M \times N$ пікселів [185, 186]:

$$\boldsymbol{I}_{M\times N} = \boldsymbol{U}_{M\times M}(\boldsymbol{I}) \times \boldsymbol{Q}_{M\times N}(\boldsymbol{I}) \times \boldsymbol{V}^{T}_{N\times N}(\boldsymbol{I}), \tag{6.4}$$

$$\boldsymbol{D}_{M\times N} = \boldsymbol{U}_{M\times M}(\boldsymbol{D}) \times \boldsymbol{Q}_{M\times N}(\boldsymbol{D}) \times \boldsymbol{V}^{T}_{N\times N}(\boldsymbol{D}), \tag{6.5}$$

$$\boldsymbol{V}_{N\times N}(\boldsymbol{S}) = \boldsymbol{V}_{N\times N}(\boldsymbol{I}) + G \times \boldsymbol{V}_{N\times N}(\boldsymbol{D}),$$

де $\boldsymbol{U}_{M\times M}(\boldsymbol{I}), \boldsymbol{V}_{N\times N}(\boldsymbol{I})$ — відповідно, матриці лівих і правих сингулярних векторів матриці $\boldsymbol{I} \times \boldsymbol{I}^{T}$ ($\boldsymbol{I}^{T} \times \boldsymbol{I}$); $\boldsymbol{Q}_{M\times N}(\boldsymbol{I}) = \boldsymbol{E}_{M\times N} \times \boldsymbol{\Lambda}(\boldsymbol{I})$ — діагональна матриця сингулярних чисел; $\boldsymbol{E}_{M\times N}$ — одинична матриця; $\boldsymbol{\Lambda}(\boldsymbol{I}) = \big\{\sqrt{\lambda_1}, \sqrt{\lambda_2}, \cdots, \sqrt{\lambda_K}\big\}$, $K = \min(M, N)$ — вектор сингулярних чисел.

Одночасна модифікація матриць лівих $\boldsymbol{U}_{M\times M}(\boldsymbol{I})$ та правих $\boldsymbol{V}_{N\times N}(\boldsymbol{I})$ сингулярних векторів — зсув рядків $\boldsymbol{u}_{i\times M}(\boldsymbol{I}), i \in [1; M]$ та стовпців $\boldsymbol{v}_{N\times j}(\boldsymbol{I}), j \in [1; N]$ на Δ_x позицій — не призводить до змін вихідного виду стеганограм $\boldsymbol{S}_{M\times N}$ [185, 186], що не дає можливості однозначно відновити СД на приймальній стороні стеганографічної системи зв'язку. Для забезпечення однозначності при виділенні СД, в [161] запропоновано модернізацію алгоритму Агарвала, згідно якої приховання повідомлень $\boldsymbol{D}_{x,y}$, представлених у вигляді півтонових ЦЗ розміром $M \times N$ пікселів, проводилося з застосуванням упорядкованого за зменшенням значень вектора сингулярних чисел $\boldsymbol{\Lambda}_{N\times 1}(\boldsymbol{I})$ зображення $\boldsymbol{I}_{x,y}$ розмірами $M \times N$ пікселів:

$$\boldsymbol{\Lambda}_{N\times 1}(\boldsymbol{S}) = \boldsymbol{\Lambda}_{N\times 1}(\boldsymbol{I}) + G \times \boldsymbol{\Lambda}_{N\times 1}(\boldsymbol{D}), \tag{6.6}$$

$$\boldsymbol{\Lambda}_{N\times 1}(\boldsymbol{D}) = \frac{1}{G} \times \big(\boldsymbol{\Lambda}_{N\times 1}(\boldsymbol{S}) - \boldsymbol{\Lambda}_{N\times 1}(\boldsymbol{I})\big). \tag{6.7}$$

Перевагою запропонованої модифікації методу Агарвала (6.6) є стійкість отримуваних стеганограм до афінних перетворень [184, 186] — повороту, масштабування (дублювання рядків/стовпців), дзеркаль-

ного відображення по горизонталі та вертикалі (зміна порядку рядків/стовпців на протилежний), а також зменшення спотворень характеристик ЦЗК, у порівнянні з випадком використання спектральних перетворень ЦЗ (дискретних косинусного та вейвлет перетворень), за рахунок приховання повідомлень з використанням найбільших сингулярних чисел λ_i, $i \in [1; K]$ ЦЗК $\boldsymbol{I}_{x,y}$, що відповідають компонентам зображення з найбільшою дисперсією (енергією) [178].

Для мінімізації спотворень кольорів ЦЗК при прихованні повідомлень в комплексних СМК [181, 182] проводиться зміна системи кольору як ЦЗК (з *RGB* на *YCbCr* або *YIQ*), так і СД, представлених у вигляді кольорових зображень (з *RGB* на *Grayscale*), згідно стандартних формул [160, 182, 187]. Для вбудовування СД використовується *Y*-складова яскравості ЦЗК (метод Елахіана) або різницева колірна *I*-складова ЦЗК (метод Ганджела).

Обробка СД $\boldsymbol{D}_{x,y}$, представлених у вигляді півтонових ЦЗ розміром $M_D \times N_D$ ($M_D = N_D$) пікселів, згідно комплексних методів Елахіана та Ганджела, проводиться з використанням прямого й зворотного перетворень Арнольда [181]:

$$\begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \mathrm{mod}\left(\boldsymbol{R}_{2\times 2} \times \begin{pmatrix} x_i \\ y_i \end{pmatrix}, M_D\right), \quad \boldsymbol{R} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}, \tag{6.8}$$

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \mathrm{mod}\left(\boldsymbol{R}_{2\times 2}^{-1} \times \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix}, M_D\right), \quad \boldsymbol{R^{-1}} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}. \tag{6.9}$$

де x, y — відповідно, позиція елемента $\boldsymbol{D}_{x,y}$ (номер рядка, стовпця); $\mathrm{mod}(a, b)$ — операція визначення залишку від ділення числа a на b; i — номер поточної ітерації перетворення Арнольда (ПА). Необхідно зазначити, що період ПА — кількість ітерацій (6.8), після яких відновлюється вихідний вигляд ЦЗ, — нелінійно залежить від розмірів СД $\boldsymbol{D}_{x,y}$ [188], що ускладнює оцінку параметрів прихованих повідомлень при проведенні ПСА.

Для оцінки стійкості стеганограм з вбудованими до ОПЦЗК даними до методів ПСА в [161] було проведено дослідження сформованих згідно одноетапних, багатоетапних та комплексних методів стеганограм при варіації в широких межах значення вагового параметра G. Оцінка змін вилучених СД $\boldsymbol{D}_{x,y}^{OUT}$ у порівнянні з приховуваним повідомленням $\boldsymbol{D}_{x,y}^{IN}$, представлених в вигляді півтонових ЦЗ, при зміні значення параметра G проводилася з використанням нормованої ВКФ $\boldsymbol{B}_{x,y}$:

$$\boldsymbol{B}_{x,y}(\boldsymbol{D}^{IN}, \boldsymbol{D}^{OUT}) = \frac{1}{Eng(\boldsymbol{D}^{IN})} \times \sum_{i=(-M_D)}^{M_D} \sum_{j=(-N_D)}^{N_D} \left\{\boldsymbol{D}_{x,y}^{IN} \times \boldsymbol{D}_{(x+i),(y+j)}^{OUT}\right\},$$

де $Eng(\boldsymbol{D}^{IN}) = \sum_{i=(-M_D)}^{M_D} \sum_{j=(-N_D)}^{N_D} \{\boldsymbol{D}_{x,y}^{IN}\}^2$ — енергія вихідних СД; M_D, N_D — розміри стеганографічних даних.

З міркувань забезпечення компромісу між робастністю стеганограм до АСА ($G \to G_{max}$) і ПСА ($G \to G_{min}$) були визначені діапазони змін вагового параметра G. Значення $G = G_{max}$ відповідає появі видимих спотворень ЦЗ при формуванні стеганограм, а $G = G_{min}$ — неможливості відновлення вихідного виду СД на приймальній стороні стеганографічної системи зв'язку: $\boldsymbol{B}_{x,y}(\boldsymbol{D}^{IN}, \boldsymbol{D}^{OUT}) \le 0.1$.

Етапи обробки ЦЗ контейнера та стеганографічних даних при формуванні стеганограм згідно одноетапних, багатоетапних та комплексних методів наведені в табл. 6.1.

Таблиця 6.1

Етапи обробки ЦЗК та СД при формуванні стеганограм за різних методів приховання повідомлень в області перетворення контейнера[XXXIV]

Етапи \ Автор методу		Дей	Агарвал	Джозеф	Хан	Елахіан	Ганджел
Перетворення системи кольорів	ЦЗ-контейнер	*RGB → RGB*				*RGB → YCbCr*	*RGB → YIQ*
	ЦЗ-СД	*RGB → RGB*				*RGB → Grayscale*	
Обробка ЦЗ-контейнера	1-й етап	ДДВП	СР	ДДВП			
	2-й етап	—	—	СР	ДДКП	—	ДДКП
	3-й етап	—	—	—	СР	—	—
Обробка ЦЗ-СД	1-й етап	ДДВП	СР	СР	СР	ПА	ПА
	2-й етап	—	—	—	—	ДДВП	—
Вагові параметр G	G_{min}	0,02		0,10	0,50	1,00	5,00
	G_{max}	0,08		2,00	4,00	12,00	14,00
	Δ_G	0,02		1,00	1,00	3,00	3,00

ПСА одноетапних, багатоетапних та комплексних методів суттєво ускладнений тим, що приховання повідомлень відбувається у різних ОПЦЗК (див. табл. 6.1). Виявлення стеганограм, сформованих згідно даних методів, потребує розробки спеціальних складних статистичних моделей ЦЗ у кожній з областей приховання повідомлень, що є нетривіальною задачею [189]. Тому для виявлення стеганограм незалежно від області приховання повідомлень найчастіше використовуються методи універсального (сліпого) стеганоаналізу [143].

[XXXIV] У таблиці прийнято наступні абревіатури: ДДВП — двовимірне дискретне вейвлет перетворення; ДДКП — двовимірне дискретне косинусне перетворення; СР — сингулярний розклад; ПА — перетворення Арнольда.

Один з найбільш відомих на сьогодні методів універсального стеганоаналізу був запропонований І. Авсібашем *(Avcibaş)* [190]. Метод Авсібаша заснований на дослідженні відмінностей результатів обробки зображень-контейнерів та стеганограм з використанням функції калібрування — НЧ-фільтрації ЦЗ із застосуванням фільтру Гауса (ФГ).

Виявлення стеганограм з використанням універсального стеганодетектора (УСД) Авсібаша проводиться в декілька етапів (рис. 6.1).

Рис. 6.1. Структурна схема блока обробки цифрових зображень з використанням універсального стеганодетектора Авсібаша

На першому етапі проводиться фільтрація заданого зображення $\boldsymbol{I}_{x,y}$ розмірами $M \times N$ пікселів з використанням ФГ [190]:

$$\boldsymbol{I}_{x,y}^{filt} = \sum_{u=(-a)}^{a} \sum_{v=(-b)}^{b} \boldsymbol{I}_{x+u,y+v} \times \boldsymbol{G}_{u,v},$$

$$\boldsymbol{G}_{x,y} = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)}, \quad x \in [-a;a], \quad y \in [-b;b],$$

$$a = (M_G - 1)/2, \quad b = (N_G - 1)/2, \qquad (6.10)$$

де $\boldsymbol{I}_{x,y}^{filt}$ — оброблене зображення; $\boldsymbol{G}_{x,y}$ — ядро Ґауса розмірами $M_G \times N_G$ пікселів; $\sigma^2 (\sigma \neq 0)$ — рівень дисперсії ФГ. Для зменшення величини граничних ефектів при застосуванні ФГ зображення $\boldsymbol{I}_{x,y}$ симетрично доповнюється $2a$ рядками та $2b$ стовпцями — (6.10).

На другому етапі роботи УСД Авсібаша, для оцінки обумовленого використанням ФГ ступеня спотворення ЦЗК $\boldsymbol{I}_{x,y}$ у просторовій і частотній областях, розраховуються стандартні метрики якості (МЯ) [190].

Отримані значення МЯ поступають до блока класифікатора на основі лінійної регресії (КЛР) $SD_{Avcibaş}$ (див. рис. 6.1). Рішення y_I щодо віднесення зображення $\boldsymbol{I}_{x,y}$ до класу незаповнених ЦЗК або стеганограм приймається згідно формули [190]:

$$y_I = \underset{i \in \{C,S\}}{\operatorname{argmin}} \left(\left| y_i - \sum_{j=1}^{L_F} \beta_j \cdot x_j \right| \right),$$

де y_C, y_S — відповідно, порогові значення для віднесення зображення до класу зображень-контейнерів або ж стеганограм; x_j — значення j-ї МЯ ЦЗ; $\boldsymbol{\beta} = (\beta_1, \beta_2 \cdots \beta_{L_F})$ — вектор ваг КЛР; L_F — кількість використаних МЯ.

Визначення вектора $\boldsymbol{\beta}$ проводиться на етапі налаштування КЛР шляхом мінімізації значення наступної цільової функції на пакеті тестових зображень та стеганограм [184]:

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{L_{T_S}} \left(y_i - \sum_{j=1}^{L_F} \beta_i \cdot x_{i,j} \right)^2, \qquad (6.111)$$

де y_i — рішення КЛР для i-го досліджуваного зображення; x_j — j-й регресор (МЯ ЦЗ); $L_F, L_{T_S}\left(L_F \le L_{T_S}\right)$ — відповідно, розмірність простору ознак та потужність множини тестових ЦЗ; $RSS(\boldsymbol{\beta})$ — сума квадратів відхилень (*Residual Sum of Squares, RSS*).

6.4. Пасивний аналіз стеганограм на основі приховання в областях перетворення цифрових зображень

Одними з основних задач при проведенні ПСА ЦЗ у випадку приховання повідомлень в областях перетворень графічних контейнерів є визначення статистичних характеристик зображень та стеганограм, налаштування статистичних та універсальних стеганодетекторів, визначення метрики якості їхньої роботи, а також оцінка ймовірності виявлення стеганограм з використанням стеганодетекторів.

Процедура налаштування статистичних та універсальних стеганодетекторів

Зважаючи на значну кількість параметрів статистичних моделей *SPAM* ($d_{SPAM} = 686$) та *CC-PEV* ($d_{CC\text{-}PEV} = 548$) ЦЗ, налаштування статистичних стеганодетекторів доцільно проводити з використанням ансамблю класифікаторів (АК) згідно методу т. зв. «випадкових лісів» (*Random forests*) [191]. Особливістю АК є використання множини базових («простих») класифікаторів (БК) для забезпечення високої точності класифікації об'єктів (цифрових зображень) у випадку використання простору ознак великої розмірності ($d_{model} \gg 100$), коли застосування поширених методів класифікації, зокрема й методу опорних векторів (*Support Vector Machine, SVM*), має суттєві обмеження [184, 192]. Рішення щодо віднесення ЦЗ до класу пустих або заповнених контейнерів (стеганограм) при використанні АК приймається за «мажори-

тарним» принципом — порівняння суми рішень окремих (базових) класифікаторів з фіксованим пороговим значенням.

Для забезпечення високої точності виявлення стеганограм при налаштуванні АК проводиться попередня обробка множини демаскуючих ознак та пакета тестових зображень [191]:

1) псевдовипадкове розбиття вихідної множини ознак стеганодетектора розмірністю d_{ALL} на декілька підмножин однакової потужності d_{sub} ($d_{sub} \ll d_{ALL}$) — для зменшення часу налаштування БК при забезпеченні фіксованої ймовірності виявлення стеганограм;

2) створення L_B вибірок тестових зображень з повтореннями згідно беггінг-методу[XXXV] — для забезпечення високої ймовірності виявлення стеганограм у випадку використання пакета тестових зображень малої потужності [184, 192].

В якості базового класифікатора АК можна, наприклад, використати лінійний дискримінант Фішера (ЛДФ), що широко застосовується при проведенні ПСА цифрових зображень [191]. З алгоритмами визначення параметрів АК (кількість L_B базових класифікаторів і розмірність d_{sub} простору ознак БК), алгоритмами налаштування АК, ЛДФ та УСД Авсібаша (визначення значень елементів вектора $\boldsymbol{\beta}$ шляхом мінімізації цільової функції (6.11)) можна ознайомитися у Додатках до [161].

Метрики якості роботи стеганодетекторів цифрових зображень

Для інтегральної оцінки ефективності стеганодетектора пропонується використати наступні метрики якості [184, 194, 195]:

1) Площа під *ROC*-кривою (*Area under ROC curve, AUC*)[XXXVI]:

$$AUC = \int_{-\infty}^{+\infty} TPR(T) \times P_0(T) dT, \quad AUC \in [0; 1], \tag{6.12}$$

де TPR *(True Positive Rate)* — параметр, що визначає кількість вірно класифікованих стеганограм; P_0 — ймовірність хибного віднесення ЦЗ до іншого класу, наприклад, віднесення пустих зображень-контейнерів до класу заповнених; T — порогове значення, що використовується для віднесення зображень до класів пустих або заповнених контейнерів;

2) ступінь впливу використовуваного простору ознак на точність детектування стеганограм (*Informedness, INs*):

[XXXV] Беггінг (*Bagging*, скор. від *Bootstrap aggregation*) — один з перших і найпростіших видів ансамблів комп'ютеризовано визначених статистик складних імовірнісних розподілів, запропонований Л. Брейманом (*Breiman*) у 1994 р. [193].

[XXXVI] *ROC (Receiver Operating Characteristic)* — робоча характеристика приймача.

$$INs = \frac{TP \times TN - FN \times FP}{(TP + FN) \times (FP + TN)}, \quad INs \in [-1; 1], \tag{6.13}$$

де TP, TN — відповідно, кількість правильно класифікованих стеганограм та зображень-контейнерів; FP, FN відповідно, кількість помилок стеганодетектора 1-го (хибне спрацювання) та 2-го (пропуск стеганограми) роду;

3) ступінь впливу використовуваного класифікатора на ефективність стеганодетектора (*Markedness*, *MNs*):

$$MNs = \frac{TP \times TN - FN \times FP}{(TP + FP) \times (TN + FN)}, \quad MNs \in [-1; 1], \tag{6.14}$$

4) коефіцієнт кореляції Меттьюза (*Matthews Correlation Coefficient*, *MCC*) — використовується для оцінки величини кореляції рішень стеганодетектора та істинних міток класів досліджуваних ЦЗ:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{6.15}$$

$$MCC \in [-1; 1].$$

Метрика AUC (6.12) широко використовується для порівняння ефективності бінарних класифікаторів [184]. Значення метрики $AUC = 0$ відповідає віднесенню ЦЗ до класу стеганограм, а $AUC = 1$ — випадку «ідеального» бінарного класифікатора (віднесення ЦЗ і стеганограм до відповідних класів із заданою точністю). Окремим випадком є значення $AUC = 0.5$, що відповідає віднесенню зображень до класів контейнерів або стеганограм випадковим чином.

Значення метрик $INs = 1$ (6.13) та $MNs = 1$ (6.14) відповідають випадку найбільш «повного» використання як простору ознак, так і класифікатора для виявлення стеганограм. При $INs = -1$ або $MNs = -1$ використання заданого простору ознак або типу класифікатора призводить до хибного віднесення ЦЗ до класу стеганограм.

Віднесення зображень до класів контейнерів або стеганограм випадковим чином відповідає значенням метрик $INs = 0$ та $MNs = 0$ — використанню ознак, що слабко змінюються при вбудовуванні СД до ЦЗ (метрика *INs*), та вибору «слабкого» класифікатора (метрика *MNs*), що не дозволяє розділяти класи ЦЗК і стеганограм з фіксованою точністю.

Первинні дані при проведені ПСА цифрових зображень

Нами була використана вибірка з $L_I = 2500$ зображень зі стандартного пакета зображень *MIRFlickr-25000* [196], широко використо-

вуваного для оцінки ефективності методів аналізу та класифікації ЦЗ провідними *IT*-компаніями, зокрема *Yahoo* [197].

Формування навчального I_{TS} ($|I_{TS}| = L_I/2$) та контрольного I_{Ev} ($|I_{Ev}| = L_I/2$) пакетів ЦЗ проводилося згідно стандартної процедури тестування стеганодетекторів [191] — шляхом псевдовипадкового обрання зображень з вибірки, з використанням програмного генератора випадкових чисел на основі алгоритму «вихор Мерсенна» (*Mersenne Twister*)[XXXVII]. У разі обрання однакових ЦЗ при формуванні навчального або тестового пакетів проводилася зміна вихідних параметрів генератора Мерсенна та повторне генерування ПВП. Для отримання усереднених значень точності детектування стеганограм, розбиття вихідного тестового пакета на навчальну I_{TS} і контрольну I_{Ev} вибірки повторювалися десять разів [191].

Стандартною практикою при проведенні ПСА ЦЗ є масштабування тестових зображень до однакового розміру, що забезпечуватиме рівну кількість даних (значень яскравості пікселів зображення) для оцінки статистичних параметрів зображень [198]. Тому при формуванні навчального та контрольного пакетів ЦЗ нами було проведено масштабування ЦЗ до однакового розміру 512×512 пікселів, що широко використовується при оцінці ефективності стеганодетекторів [157, 162]. Масштабування ЦЗ проводилося з використанням стандартного методу — бікубічної інтерполяції (*bicubic interpolation*) [160].

В якості приховуваних повідомлень були використані повнокольорові ЦЗ з різним ступенем деталізації (рис. 6.2) — креслення (*а*), карта (*б*) та портрет (*в*). Параметри тестових ЦЗ та стеганографічних даних наведені в табл. 6.2.

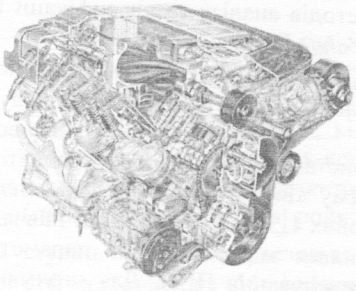
Для оцінки ефективності роботи стеганодетектора при варіації об'єму СД, вбудованих до зображення-контейнера, нами проводилася зміна ступеня заповнення ЦЗК стеганографічними даними Δ_C:

$$\Delta_C = \left(L_{changed}/L_{all}\right) \times 100\%.$$

де $L_{changed}$, L_{all} — відповідно, частка змінених та загальна кількість коефіцієнтів розкладу зображення-контейнера в заданому базисі.

Ступінь заповнення Δ_C ЦЗК стеганоданими змінювалася від 5% до 25%, з кроком 5%, та від 25% до 95%, з кроком 10%. Вибір меншого кроку зміни Δ_C в діапазоні від 5% до 25% був зумовлений тим, що зазначений діапазон відповідає випадку слабкого заповнення ЦЗК даними, і саме це представляє особливий інтерес при проведенні ПСА [162, 199].

[XXXVII] Вихор Мерсенна — генератор високоякісних за критерієм випадковості ПВЧ, розроблений у 1997 р. яп. вченими Макото Мацумото (*Makoto Matsumoto*) і Такудзі Нішімура (*Takuji Nishimura*). Базується на властивостях простих чисел Мерсенна.



а)

б)

в)

Рис. 6.2. Приклади СД типу: *(а)* – креслення; *(б)* – карта; *(в)* – портрет

Таблиця 6.2

Характеристики тестових цифрових зображень та стеганографічних даних

Параметр	Тестові ЦЗ	СД		
		креслення	карта	портрет
Роздільна здатність, пікс.	512×512	567×463	800×800	565×850
Система кольорів	*RGB*	*RGB*	*RGB*	*RGB*
Глибина кольору, біт/пікс.	24	24	24	24
Формат представлення	*JPEG, TrueColor*	*BMP*	*BMP*	*BMP*

Формування стеганограм з даними, вбудованими до ОПЦЗК згідно одноетапних методів Дея та Агарвала, багатоетапних методів Джозефа та Хана, а також комплексних методів Елахіана та Ґанджела, проводилося за алгоритмами, детальний опис яких наведено в Додатку [161]. Згідно рекомендацій [177, 179–182], в якості базисних функцій двовимірного ДВП у методах Дея, Джозефа, Хана, Елахіана та Ґанджела було використано вейвлет Гаара (*Haar wavelet*) та відповідна йому скейлінг-функція. Значення вагового параметра G при формуванні стеганограм згідно кожного методу приховання повідомлень змінювалися в діапазоні від G_{min} до G_{max} з кроком Δ_G (див. табл. 6.1). Кількість ітерацій ПА для комплексних методів Елахіана [181] та Ґанджела [182] була обрана рівною $N_A = 23$, що не є дільником розмірів СД (табл. 6.2) та, відповідно, не призводить до зменшення періоду ПА [188].

Визначення стійкості стеганографічних методів приховання повідомлень до ОПЦЗК щодо відомих алгоритмів ПСА потребує проведення досліджень імовірності виявлення стеганограм за різних способів налаштування стеганокодера — приховання різних типів повідомлень, наприклад, СД типу «Креслення», «Карта» та «Портрет» (табл. 6.2), використання всіх/окремих каналів кольору ЦЗК. Для прикладу, наведемо результати дослідження точності виявлення (метрика AUC) стеганограм, сформованих згідно методу Дея, при вбудовуванні СД типу «Креслення», «Карта» та «Портрет» до різних каналів кольору ЦЗК. Значення метрики AUC при застосуванні стеганодетектора SD_{SPAM} для виявлення сформованих стеганограм у випадку слабкого заповнення графічного контейнера стеганоданими ($\Delta_C = 5\%$) та малих значень параметра G, наведені в табл. 6.3.

При вбудовуванні СД з використанням каналів червоного та синього кольору ЦЗК спостерігається зниження ймовірності виявлення прихованих повідомлень (метрика AUC) у порівнянні з випадком використання каналу зеленого кольору (див. табл. 6.3). Це пояснюється нерівномірним розподілом енергії шумів (наприклад, теплового та дробового шуму елементів матриці фоточутливих елементів) по окремих каналах кольору ЦЗ внаслідок використання спеціальних фільтрів, зокрема фільтра Баєра[XXXVIII] [160, 200], для розділення кольорових складових зображення на етапі формування ЦЗ [143].

[XXXVIII] Фільтр Баєра (або т. зв. шаблон Баєра) — це масив кольорових *RGB*-фільтрів у комірках матриці, якими вкрито фотодіоди, використовувані у цифровій фотографії. Названо на честь Брайса Баєра (*Bryce Bayer*), співробітника компанії *Kodak*. Фільтр використовують у фотосенсорах цифрових фотоапаратів, відеокамер і сканерів для одержання кольорового зображення. Масив фільтра складається з 25% червоних елементів, 25% синіх і 50% зелених елементів, через що його часто називають *GRGB* чи *RGBG*-фільтром.

Таблиця 6.3

Метрики *AUC* при застосуванні стеганодетектора SD_{SPAM} для виявлення стеганограм, сформованих згідно методу Дея з використанням різних каналів кольору контейнера та типів СД

Тип СД	Канал червоного кольору	Канал зеленого кольору	Канал синього кольору
Усі типи СД	0.817	0.890	0.821
Креслення	0.806	0.883	0.802
Карта	0.799	0.876	0.793
Портрет	0.793	0.873	0.791

Вбудовування різних типів СД до ОПЦЗК не призводить до суттєвих змін точності детектування стеганограм (метрика *AUC*, табл. 6.3). Зростання ймовірності виявлення прихованих повідомлень спостерігається лише у випадку використання всіх типів СД при налаштуванні стеганодетектора, що обумовлено врахуванням особливостей прихованих повідомлень при проведенні ПСА — розподілу енергії СД по окремих частотних піддіапазонах, кореляційних характеристик повідомлень.

Використання апріорних даних щодо характеристик СД — наприклад, розподілу енергії приховуваних повідомлень в різних частотних піддіапазонах, а також кореляції між каналами кольору ЦЗК при налаштуванні стеганодетектора — дозволяє підвищити точність виявлення стеганограм. Проте, в загальному випадку проведення ПСА зазначені дані є відсутніми, тому визначення стійкості одноетапних, багатоетапних та комплексних методів приховання повідомлень до ОПЦЗК проводитимемо для найбільш складного випадку, для якого досягаються найменші значення ймовірності виявлення СД (табл. 6.3) — приховання повідомлень типу «Портрет» до каналу синього кольору ЦЗК. Зі значеннями метрики *AUC* при проведенні ПСА ЦЗ із застосуванням статистичних та універсальних стеганодетекторів та вбудовуванні СД типу «Креслення» й «Карта» до каналів червоного та зеленого кольору ЗК можна ознайомитися у Додатках [161].

6.4.1. Виявлення стеганограм з використанням методів статистичного стеганоаналізу

Для оцінки ймовірності виявлення стеганограм з даними, вбудованими до ОПЦЗК згідно одноетапних, багатоетапних і комплексних методів, нами були використані стеганодетектори SD_{SPAM} та $SD_{CC\text{-}PEV}$, налаштовані, відповідно, із застосуванням статистичних моделей ЦЗ у просторовій (модель *SPAM*) та частотній (модель *CC-PEV*) областях. Сімейства залежностей значень метрики *AUC* від ступеня заповнення графічного контейнера стеганоданими Δ_C та вагового коефіцієнта G для

стеганограм з СД, вбудованими до області перетворення контейнера, при використанні стеганодетекторів SD_{SPAM} і $SD_{CC\text{-}PEV}$ ($IQF = 100$) наведені на рис. 6.3, 6.4.

Використання статистичної моделі *SPAM* дозволяє з високою точністю ($AUC > 0.99$) виявляти стеганограми, сформовані згідно багатоетапного методу Хана (рис. 6.3, *г*) та комплексних методів Елахіана (рис. 6.3, *д*) та Ганджела (рис. 6.3, *е*), незалежно від ступеня заповнення ЦЗК стеганоданими та значення вагового параметра G. Отриманий результат є неочікуваним, враховуючи, що дані методи вбудовування СД до ОПЦЗК були запропоновані саме з метою підвищення стійкості стеганограм до методів статистичного стеганоаналізу. Відносно низька стійкість стеганограм, сформованих згідно зазначених методів, до статистичного стеганоаналізу пояснюється суттєвим зменшенням значень параметрів моделі *SPAM* — величини кореляції різниць значень яскравості суміжних пікселів — в областях зображення, використаних при вбудовуванні СД, у порівнянні зі значеннями параметрів моделі *SPAM* для графічного контейнера.

При використанні одноетапних методів Дея (рис. 6.3, *а*) та Агарвала (рис. 6.3, *б*) значення AUC суттєво зменшується при слабкому заповненні контейнера СД ($\Delta_C \leq 10\%$) та мінімальних значеннях вагового параметра G. Це пояснюється незначними змінами значень кореляції яскравостей суміжних пікселів ЦЗ при вбудовуванні СД з використанням спектральних і спеціальних перетворень ЦЗК, що не виявляються при використанні статистичної моделі *SPAM*.

Найменші значення метрики AUC для стеганодетектора SD_{SPAM} досягаються у випадку використання двоетапного методу Джозефа — $AUC_{min} = 0.585$ (рис. 6.3, *в*), — що практично відповідає віднесенню ЦЗ до класів зображень-контейнерів чи стеганограм випадковим чином. Суттєве зниження точності виявлення стеганограм, сформованих при використанні як спектральних, так і спеціальних перетворень ЦЗК (метод Джозефа), пояснюється приховуванням повідомлень на рівні власних шумів контейнера.

ПСА зображень на основі статистичної моделі *CC-PEV* має низьку ймовірність виявлення стеганограм, сформованих із застосуванням спектральних перетворень ЦЗК — одноетапного методу Дея та двоетапного методу Джозефа, — особливо в області слабкого заповнення контейнера стеганоданими ($\Delta_C < 10\%$, рис. 6.4). Виявлене зниження точності детектування стеганограм пояснюється тим, що коефіцієнти двовимірного ДКП ЦЗ отримуються лише для невеликої частини останнього (блока розбиття), в межах якої зміни статистичних характеристик ЦЗК, унаслідок вбудовування СД до ОПЦЗК, є відносно невеликими.

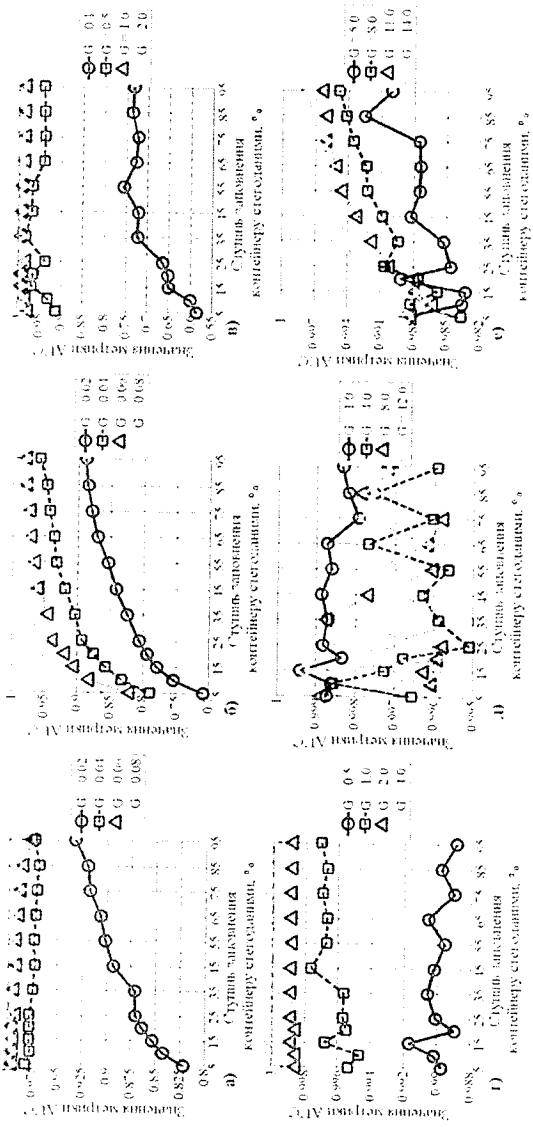


Рис. 6.3. Залежності значень метрики *AUC* від ступеня заповнення контейнера СД при варіації вагового параметра *G* та використанні стеганодетектора SD_{SPAM} для виявлення стеганограм, сформованих згідно: *(а)* – методу Дея; *(б)* – методу Агарвала; *(в)* – методу Джозефа; *(г)* – методу Хана; *(д)* – методу Елахіана; *(е)* – методу Ганджела

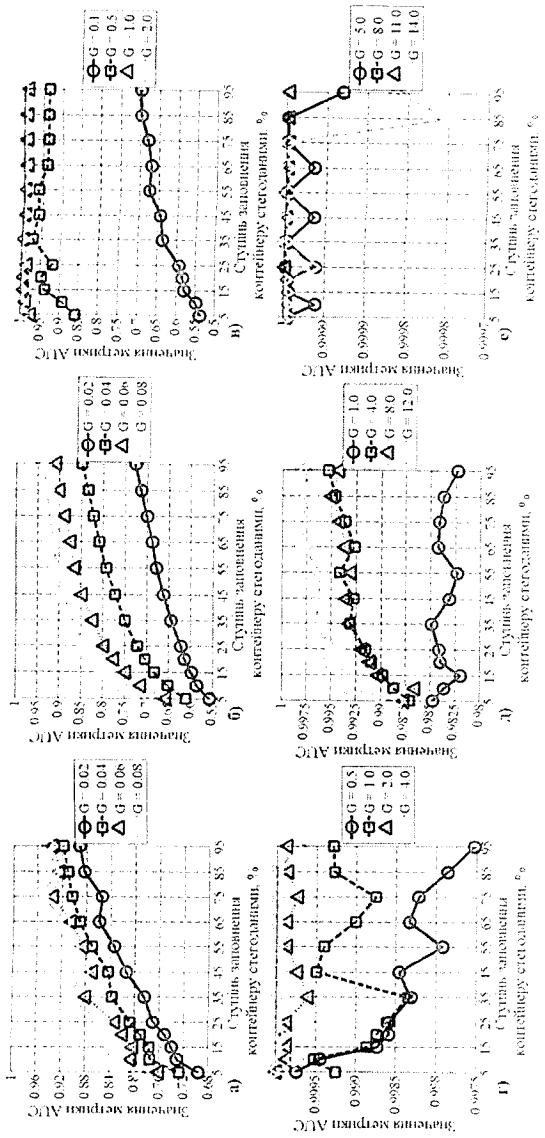


Рис. 6.4. Залежності значень метрики *AUC* від ступеня заповнення контейнера СД при варіації вагового параметра *G* та використанні стеганодетектора $SD_{CC\text{-}PEV}$ (*IQF* = 100) для виявлення стеганограм, сформованих згідно: *(а)* – методу Дея; *(б)* – методу Агарвала; *(в)* – методу Джозефа; *(г)* – методу Хана; *(д)* – методу Елахіана; *(е)* – методу Ганджела

Мінімальні зміни значень коефіцієнтів двовимірного ДКП у суміжних блоках розбиття ЦЗ, обумовлені приховуванням повідомлень згідно методів Дея та Джозефа, обмежують використання статистичних характеристик коефіцієнтів двовимірного ДКП між блоками розбиття зображень для виявлення стеганограм.

Враховуючи зниження точності виявлення стеганограм, сформованих згідно одноетапних методів Дея та Агарвала, а також двоетапного методу Джозефа, при застосуванні статистичних моделей *SPAM* та *CC-PEV*, становить інтерес використання універсальної статистичної моделі для підвищення ймовірності детектування стеганограм. Саме такою є модель *CDF* (*Cross-Domain Feature*), сформована на основі об'єднання основних елементів моделей *SPAM* ($d_{SPAM} = 686$ параметрів) і *CC-PEV* ($d_{CC\text{-}PEV} = 548$ параметрів) [201]. Перевагою використання моделі *CDF* є можливість виявлення стеганограм незалежно від області приховання повідомлень. Розмірність моделі *CDF*: $d_{CDF} = d_{SPAM} + d_{CC\text{-}PEV} = 1234$ параметрів.

Залежності ймовірності виявлення стеганограм (метрика *AUC*) від ступеня заповнення ЦЗК стеганоданими при варіації вагового параметра G і використанні стеганодетектора SD_{CDF} на основі статистичної моделі *CDF* для виявлення стеганограм з даними, вбудованими до ОПЦЗК згідно одноетапних, багатоетапних і комплексних методів, наведені на рис. 6.5.

Незважаючи на суттєве зростання розмірності простору ознак стеганодетектора при використанні універсальної статистичної моделі *CDF* ($d_{CDF} = 1234$) у порівнянні з моделями *SPAM* ($d_{SPAM} = 686$) та *CC-PEV* ($d_{CC\text{-}PEV} = 548$), відповідне зростання значення метрики *AUC* (табл. 6.4) є відносно малим — $\Delta_{AUC} \leq 0.055$ (рис. 6.5).

Для порівняння, у табл. 6.4 наведено значення ймовірності виявлення стеганограм (метрика *AUC*), сформованих згідно одноетапних методів Дея та Агарвала, багатоетапних методів Джозефа та Хана, а також комплексних методів Елахіана та Ганджела, при використанні статистичних моделей зображення-контейнера *SPAM*, *CC-PEV* та *CDF* у випадку слабкого заповнення контейнера стеганоданими ($\Delta_C < 10\%$) та за мінімальної енергії повідомлень ($G = G_{min}$).

Зменшення параметра *IQF*, що відповідає збільшенню ступеня *JPEG*-стиснення ЦЗ, при використанні статистичної моделі *CC-PEV* (табл. 6.4) призводить до зниження точності детектування стеганограм, сформованих згідно одноетапних методів Дея та Агарвала, багатоетапного методу Хана та комплексних методів Елахіана і Ганджела. Це обумовлено тим, що вбудовування СД до ЦЗ згідно зазначених методів проводиться з використанням компонентів контейнера з найбільшою енергією — апроксимуючих коефіцієнтів двовимірного ДВП, найбільших сингулярних чисел контейнера, — що відповідають НЧ-коефіцієнтам двовимірного ДКП, а отже несуттєво змінюються при зміні ступеня *JPEG*-стиснення графічного контейнера [160, 121].

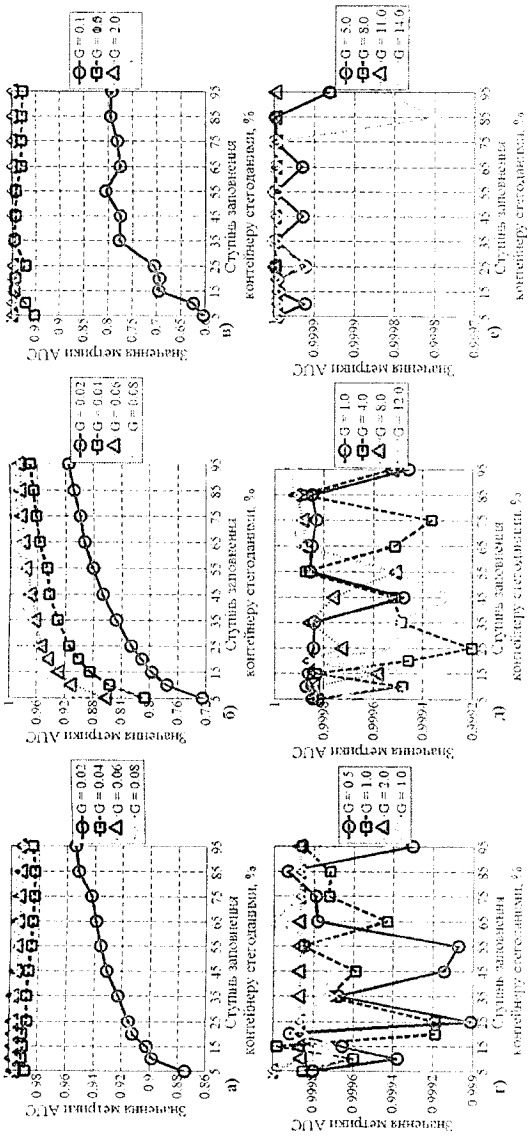


Рис. 6.5. Залежності значень метрики *AUC* від ступеня заповнення контейнера СД при варіації вагового параметра *G* при використанні статистичної моделі *CDF* для виявлення стеганограм, сформованих згідно:
(а) – методу Дея; *(б)* – методу Агарвала; *(в)* – методу Джозефа;
(г) – методу Хана; *(д)* – методу Елахіана; *(е)* – методу Ганджела

Таблиця 6.4

Метрики AUC при застосуванні стеганодетекторів SD_{SPAM}, $SD_{CC\text{-}PEV}$ та SD_{CDF} для виявлення стеганограм, сформованих згідно одноетапних, багатоетапних та комплексних методів, при слабкому заповненні контейнера СД ($\Delta_C = 10\%$) та малій енергії прихованих повідомлень

	Статистична модель зображення-контейнера			
	SPAM	*CC-PEV* ($JQF = 90$)	*CC-PEV* ($JQF = 100$)	*CDF*
Метод Дея	0.843	0.710	0.730	0.898
Метод Агарвала	0.753	0.542	0.586	0.775
Метод Джозефа	0.585	0.569	0.549	0.623
Метод Хана	0.990	0.932	0.999	0.999
Метод Елахіана	0.999	0.622	0.984	0.999
Метод Ганджела	0.984	0.999	0.999	0.999

Таким чином, можна зробити висновок, що використання відомих статистичних моделей ЦЗ не дозволяє виявляти з високою точністю стеганограми, сформовані згідно одноетапних методів Дея та Агарвала, а також двоетапного методу Джозефа, особливо у випадку слабкого заповнення ЦЗК стеганоданими ($\Delta_C \leq 10\%$) та мінімальних значень вагового параметра G. Зниження ефективності статистичних моделей ЦЗ у випадку використання одноетапних методів приховання повідомлень пояснюється відносно малими змінами величини кореляції як яскравості суміжних пікселів графічного контейнера, так і кореляції значень коефіцієнтів двовимірного ДКП всередині та між блоками розбиття ЦЗ. Тому представляє інтерес використання методів універсального стеганодетектування, зокрема УСД Авсібаша, для виявлення стеганограм незалежно від типу перетворень ЦЗК, використаних для вбудовування СД.

6.4.2. Виявлення стеганограм з використанням універсального стеганодетектора Авсібаша

Для підвищення ймовірності виявлення стеганограм з даними, прихованими з використанням спектральних перетворень графічного контейнера, згідно рекомендацій [202–206], була проведена модифікація УСД Авсібаша — розширення переліку метрик якості ЦЗ групою метрик візуальної якості зображень. Унаслідок цього загальна кількість МЯ $d_{Avcibaş}$, що використовувалися при налаштуванні УСД Авсібаша, стала рівною чотирнадцяти. Алгоритм обробки ЦЗ з використанням модифікованого стеганодетектора Авсібаша наведений у Додатку до [161].

Приховання повідомлень з використанням спеціальних перетворень графічного контейнера, зокрема сингулярного перетворення для методів Агарвала та Джозефа, призводить до перерозподілу енергії між компо-

нентами ЦЗК, що відповідають різним сингулярним числам матриць яскравості останнього (див. (6.6)), та, відповідно, підсилення його власних шумів. Тому перспективним є використання стандартних методів обробки зображень — медіанної [160, 183] та вінерівської [160] фільтрації ЦЗ — для придушення зазначених шумів та підвищення точності виявлення стеганограм, сформованих з використанням спеціальних перетворень графічних контейнерів.

Значення метрики AUC для випадків слабкого ($\Delta_C = 10\%,\ G = G_{min}$) і сильного ($\Delta_C = 85\%,\ G = G_{max}$) заповнення ЦЗК стеганоданими при використанні УСД Авсібаша для виявлення стеганограм з даними, вбудованими до ОПЦЗК згідно одноетапних, багатоетапних і комплексних методів, наведені на рис. 6.6.

Найбільший вплив на ймовірність виявлення стеганограм (метрика AUC, рис. 6.6) мають кореляційні характеристики зображень та метрики візуальної якості ЦЗ. Це підтверджує зроблений раніше висновок, що приховання повідомлень до ОПЦЗК призводить до суттєвих змін кореляційних характеристик останнього і параметрів шумових компонент.

У разі сильного заповнення ЦЗК СД ($\Delta_C = 85\%,\ G = G_{max}$) використання ґаусівської фільтрації ЦЗ та всіх груп МЯ зображень при налаштуванні УСД Авсібаша дозволяє з високою точністю виявляти стеганограми, сформовані з використання багатоетапного методу Хана ($AUC = 0.999$) та комплексного методу Ганджела ($AUC = 0.943$). Проте ефективність УСД Авсібаша суттєво знижується у випадку формування стеганограм згідно одноетапного методу Агарвала ($AUC = 0.646$) та комплексного методу Елахіана ($AUC = 0.720$) (рис. 6.6).

Застосування для обробки досліджуваних ЦЗ медіанної (рис. 6.6, *д*) та вінерівської (рис. 6.6, *е*) фільтрації дає можливість дещо збільшити точність виявлення стеганограм порівняно з випадком використання фільтру Ґауса (рис. 6.6, *г*) при вбудовуванні СД згідно методу Агарвала ($\Delta AUC \cong 0.052$), Джозефа ($\Delta AUC \cong 0.037$) й Ганджела ($\Delta AUC \cong 0.141$) при сильному заповненні ЦЗК стеганоданими ($\Delta_C = 85\%$).

У випадку слабкого заповнення контейнера СД (рис. 6.6, *а*) значення метрики AUC для універсального стеганодетектора Авсібаша є суттєво меншим за відповідні результати для статистичних моделей ЦЗ, зокрема моделі CDF, при формуванні стеганограм згідно одноетапних та комплексних стеганографічних методів (табл. 6.5).

Вагомим обмеженням використання як статистичних моделей ЦЗК, так і УСД Авсібаша є низька ймовірність виявлення стеганограм (метрика AUC), сформованих згідно методів Дея, Агарвала та Джозефа (табл. 6.5). Застосування спектральних (двовимірних ДВП і ДКП) і спеціальних (сингулярний розклад) методів обробки графічних контейнера та СД мінімально змінює кореляційні характеристики як ЦЗК, так

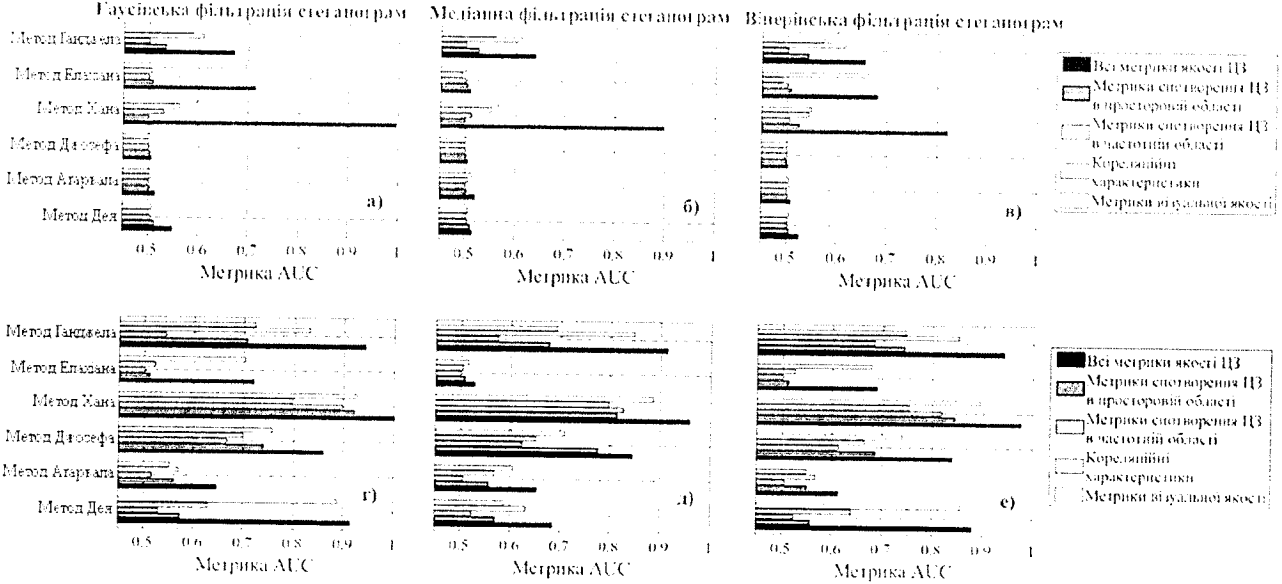


Рис. 6.6. Значення метрики *AUC* для стеганодетектора Авсібаша при слабкому *(а-в)* та сильному *(г-д)* заповненні ЦЗК стеганоданими та використанні для обробки стеганограм: *(а, г)* – фільтра Гауса; *(б, д)* – медіанного фільтра: *(в, е)* – фільтра Вінера. Стеганограми сформовані згідно одноетапних методів Дея та Агарвала, багатоетапного методу Хана та комплексних методів Елахіана і Ганджела

Таблиця 6.5

Метрики AUC при застосуванні стеганодетектора SD_{CDF} та УСД Авсібаша (використанні всіх груп МЯ та обробці ЦЗ ґаусівською, медіанною та вінерівською фільтрацією) для виявлення стеганограм, сформованих згідно одностапних, багатостапних та комплексних методів, при слабкому заповненні контейнера СД ($\Delta_C = 10\%$) та мінімальній енергії прихованих повідомлень

	Статистична модель *CDF*	Метод обробки зображень в УСД Авсібаша:		
		фільтр Гауса	медіанний фільтр	фільтр Вінера
Метод Дея	0.898	0.551	0.515	0.526
Метод Аґарвала	0.775	0.514	0.520	0.510
Метод Джозефа	0.623	0.507	0.505	0.504
Метод Хана	0.999	0.997	0.897	0.821
Метод Елахіана	0.999	0.713	0.510	0.680
Метод Ґанджела	0.999	0.671	0.640	0.655

так і його окремих компонентів, що потребує пошуку нових методів аналізу зображень, які дозволять з високою точністю виявляти навіть слабкі зміни кореляційних характеристик окремих компонент ЦЗ, обумовлених приховуванням повідомлень до ОПЦЗК.

6.5. Перспективні методи пасивного стеганоаналізу цифрових зображень

Одним з методів підвищення точності оцінки кореляційних характеристик ЦЗ є використання теорії марківських випадкових полів [207], зокрема методу імітування «відпалення» (*annealing*). Вагомим обмеженням практичного застосування даного підходу є необхідність попереднього аналізу зображення з метою визначення оптимальних розмірів околів для моделювання кореляції яскравості суміжних пікселів, що потребує використання обчислювально складних методів [207].

Пошук оптимальних методів виділення шумових компонент ЦЗ для подальшого статистичного моделювання в задачах ПСА є нетривіальною задачею, що потребує дослідження значної кількості відомих методів обробки ЦЗ, наприклад, алгоритмів нелінійної фільтрації [160, 183], використання методу головних компонент [184], застосування двовимірного ДВП з анізотропними базисними функціями [208] — *ridgelets*, *curvelets*, *bandlets* тощо. Унаслідок цього розробка нових методів статистичного стеганоаналізу ЦЗ для виявлення стеганограм з даними, вбудованими до ОПЦЗК, є нетривіальною та обчислювально складною задачею.

При проведенні ПСА одним з найбільш відомих підходів у дослідженні характеристик областей зображень є використання методів текстурного аналізу [209]. Незважаючи на відсутність узагальненого визначення поняття «текстура» [210], значна кількість дослідників [212] звертає особливу увагу на «багаторівневість» представлення текстур — наприклад, з використанням контурів об'єктів на зображенні, їхнього взаємного розташування та інформації про кольорову складову текстури. Урахування зазначених даних при формуванні стеганограм дозволяє адаптивно обирати області ЦЗК для вбудовування СД та, відповідно, мінімізувати спотворення статистичних параметрів контейнера.

Враховуючи зазначені обмеження відомих методів аналізу зображень, становить інтерес розробка багаторівневої моделі представлення ЦЗ для виявлення змін характеристик як ЦЗК загалом, так і його окремих складових (компонент) при прихованні повідомлень. Для дослідження характеристик окремих компонент ЦЗ перспективним є використання новітніх методів обробки сигналів, зокрема структурного аналізу.

Структурний аналіз сигналів сьогодні широко використовується при розв'язанні значної кількості прикладних задач: дослідження кореляційних характеристик сигналів [213], аналізу внутрішньої структури сигналів [214, 215], інтерполяції [216] та моделювання [217–219] багатовимірних даних, за умови обмеженості апріорних даних щодо їх статистичних характеристик, декомпозиції сигналів з урахуванням їх статистичних особливостей [186, 220, 221] тощо. Враховуючи, що використання спектральних та спеціальних перетворень стеганографічних даних і зображень-контейнерів змінює кореляційні характеристики останніх мінімально, становить інтерес використання спеціальних методів структурного аналізу, зокрема варіограмного та флуктуаційного аналізів, на основі яких можливе виявлення слабких змін кореляційних характеристик графічних контейнерів, обумовлених прихованням повідомлень до ОПЦЗК. Для виявлення змін фрактальних характеристик окремих компонент ЦЗК, зумовлених прихованням повідомлень до областей перетворень останніх із застосуванням сингулярного розкладу каналів кольору зображення, перспективним бачиться й використання методів мультифрактального аналізу.

6.6. Структурний стеганоаналіз цифрових зображень

Використання статистичних та універсальних стеганодетекторів дозволяє з високою ймовірністю виявляти стеганограми, сформовані згідно багатоетапного методу Хана, а також комплексних методів Елахіана та Ганджела. Проте ефективність таких детекторів суттєво знижується у

випадку приховання повідомлень до ОПЦЗК згідно одноетапних методів Дея і Агарвала, а також двоетапного методу Джозефа — особливо при слабкому заповненні контейнера стеганоданими ($\Delta_C \leq 10\%$) та мінімальних значеннях енергії приховуваних повідомлень. Зниження ймовірності виявлення одноетапних і двоетапних методів вбудовування СД до ОПЦЗК пояснюється суттєвим зменшенням змін кореляційних характеристик останнього при прихованні повідомлень з використанням декількох його складових, що відповідають різним сингулярним числам (6.6). Це потребує створення багаторівневої моделі представлення ЦЗ для виявлення змін характеристик як зображення-контейнера загалом, так і його окремих складових (компонент) при прихованні повідомлень, а також розробки методів вибору, оцінки й використання характеристик моделі у задачах ПСА.

Для виявлення слабких змін кореляційних характеристик ЦЗК, обумовлених прихованням повідомлень до області перетворень останніх, *перспективним є використання методів структурного аналізу ЦЗ. Методи* варіограмного [213, 214] та флуктуаційного [222] аналізів, що широко використовуються для досліджень кореляційних характеристик складних сигналів різної природи, дозволять виявити навіть слабкі відмінності між зображеннями-контейнерами та стеганограмами, одержаними внаслідок вбудовування СД до ОПЦЗК. Для виявлення слабких змін фрактальних параметрів окремих компонент ЦЗК, обумовлених прихованням повідомлень, представляє інтерес використання мультифрактального аналізу (МФА) [220, 221], що дозволяє проводити адаптивну декомпозицію складних сигналів (стеганограм) з урахуванням особливостей їх статистичних характеристик.

6.6.1. Багаторівнева модель цифрових зображень

Одним з сучасних підходів до моделювання багатовимірних даних, зокрема цифрових зображень, є застосування ієрархічних статистичних моделей (ІСМ) — представлення заданого сигналу у вигляді композиції реалізацій декількох стохастичних процесів [213, 214, 217, 223]. Зважаючи на високу обчислювальну складність відомих методів оцінки параметрів зазначених процесів, при проведенні ПСА ЦЗ у більшості випадків використовується спрощена ІСМ зображень, що враховує лише власні шуми зображення-контейнера [162]. Це дозволяє з високою точністю виявляти стеганограми з даними, вбудованими згідно поширених стеганографічних методів, проте суттєво знижує ефективність методів ПСА у випадку приховання повідомлень до області ОПЦЗК з використанням спеціальних (наприклад, сингулярний розклад) перетворень останнього, або ж композиції декількох перетворень.

Для підвищення точності виявлення стеганограм з даними, вбудованими до ОПЦЗК, нами пропонується багаторівнева модель представлення ЦЗ, що відрізняється від поширених моделей ЦЗК [162, 207, 208] урахуванням не лише власних шумів ε^I, а й контурів Y^C і текстур Y^T об'єктів на ЦЗК. Це дає можливість представити процес формування стеганограм $S(\boldsymbol{s})$, як внесення спотворень $\varepsilon^D(\boldsymbol{s})$ до декількох компонент контейнера-оригіналу $I(\boldsymbol{s})$:

$$S(\boldsymbol{s}) = I(\boldsymbol{s}) + \varepsilon^D(\boldsymbol{s}) = \sum_i Y_i^C(\boldsymbol{s}) + \sum_j Y_j^T(\boldsymbol{s}) + \sum_k \varepsilon_k^I(\boldsymbol{s}) + \sum_l \varepsilon_l^D(\boldsymbol{s}), \quad (6.166)$$

де $Y(\boldsymbol{s}) = \sum_* Y(\boldsymbol{s})$ — компоненти ЦЗК, що відповідають значенням яскравості пікселів у текстурних областях (Y^T) і контурах об'єктів (Y^C) на зображенні; $\varepsilon^I = \sum_k \varepsilon_k^I(\boldsymbol{s})$ — суперпозиція теплового й дробового шумів матриці фоточутливих елементів, шумів квантування та обробки ЦЗ; D_s — область матриці фоточутливих елементів, що використовуються при формуванні ЦЗ $I(\boldsymbol{s})$; $\boldsymbol{s}$ ($\boldsymbol{s} = \{s_i,\ i \in [1, n]\}$) — позиції пікселів ЦЗ у межах D_s.

Для спрощення статистичного аналізу моделі (6.166) приймемо стандартні припущення щодо просторової та внутрішньої стаціонарності статистичних характеристик зображення [213, 214]:

1) просторова стаціонарність статистичних характеристик ЦЗ досягається за виконання наступних умов:

а) однорідність сигналу — шумові компоненти $\varepsilon_k(\boldsymbol{s})$ мають однакові розподіли для кожного пікселя s_i ($s_i \in D_s,\ i \in [1, n]$);

б) ізотропія статистичних характеристик сигналу — сумісні розподіли шумових компонентів $\varepsilon_i^I(\boldsymbol{s})$ та $\varepsilon_j^I(\boldsymbol{s})$ залежать лише від відстані між пікселями;

2) внутрішня стаціонарність статистичних характеристик ЦЗ — значення коваріації яскравості пікселів зображення

$$\operatorname{cov}\big(I(\boldsymbol{v}_N), I(\boldsymbol{u}_N)\big) = C(h)$$

залежить лише від позиції пікселів $\boldsymbol{v}_N, \boldsymbol{u}_N$: $|v_i - u_i| = h, i \in [1; N]$;

3) стаціонарність функції коваріації — с окремим випадком внутрішньої стаціонарності статистичних характеристик сигналу $I(\boldsymbol{s})$ (значень яскравості пікселів окремих каналів кольору ЦЗК) та забезпечується виконанням наступних умов:

$$E[I(\boldsymbol{s})] = \mu \ \ (\mu = const),$$
$$\operatorname{cov}\big(I(\boldsymbol{v}_N), I(\boldsymbol{u}_N)\big) = C(h) \ \ (\boldsymbol{v}_N, \boldsymbol{u}_N: |v_i - u_i| = h, i \in [1; N]),$$

де $E[\cdot]$ — оператор усереднення.

Зазначені припущення є справедливими для зображення-контейнера у випадку відсутності в ньому прихованих повідомлень. Вбудовування СД порушує стаціонарність статистичних характеристик ЦЗК, зокрема значень коваріації $\mathrm{cov}\big(I(\boldsymbol{v}_N), I(\boldsymbol{u}_N)\big)$ яскравості пікселів ЦЗ, що може бути використаним для виявлення сформованих стеганограм $S(\boldsymbol{s})$.

6.6.2. Визначення характеристик багаторівневої моделі цифрових зображень

6.6.2.1. Варіограмний аналіз цифрових зображень

Для виявлення змін значень коваріації яскравості пікселів ЦЗ, обумовлених приховуванням повідомлень, представляє інтерес дослідження величини т. зв. *nugget*-ефекту (ефекту самородка) [213, 214] — ступеня варіативності значень яскравості суміжних пікселів $I(\boldsymbol{s})$:

$$\mathrm{cov}\big(I(\boldsymbol{s}), I(\boldsymbol{s}+h)\big) = C_I(h) = \begin{cases} \sigma_Y^2 + \sigma_\varepsilon^2, & h \neq 0; \\ C_Y(h), & h = 0, \end{cases}$$

$$\lim_{h\to 0}\big(C_I(0) - C_I(h)\big) \equiv C_I(0_+) = \sigma_0^2 + \sigma_\varepsilon^2 = N_I \geq 0, \qquad (6.17)$$

де $\sigma_Y^2 = C_Y(0_+) \equiv \lim\limits_{h\to 0}\big(C_Y(0) - C_Y(h)\big)$ — коваріація значень яскравості суміжних пікселів компонент $Y(\boldsymbol{s})$ зображення $I(\boldsymbol{s})$; $\sigma_\varepsilon^2 = \sigma_{\varepsilon_I}^2 + \sigma_{\varepsilon_D}^2$ — дисперсія шумів окремих каналів кольору ЦЗ, що включає власні шуми зображення $\varepsilon_I(\boldsymbol{s})$ й шуми $\varepsilon_D(\boldsymbol{s})$, зумовлені приховуванням повідомлень; h — відстань (лаг) між пікселями ЦЗ. Для забезпечення коректного (однозначного) відновлення прихованих повідомлень на приймальному боці стеганографічної системи зв'язку, енергія (дисперсія $\sigma_{\varepsilon_D}^2$) СД не повинна бути меншою за енергію $\sigma_{\varepsilon_I}^2$ власних шумів ЦЗК: $\sigma_{\varepsilon_I}^2 \leq \sigma_{\varepsilon_D}^2$.

Оцінка величини *nugget*-ефекту N_I (6.17) зображень-контейнерів $I(\boldsymbol{s})$ потребує обробки статистичних даних — дисперсії значень яскравості пікселів структурних компонентів $Y(\boldsymbol{s})$ і власних шумів $\varepsilon(\boldsymbol{s})$ ЦЗ — з пакета тестових зображень великої потужності (що за своєю сутністю є нетривіальною задачею). Для оцінки N_I нами використовуватиметься варіограмний аналіз (ВА) [213, 214], заснований на дослідженні функції-варіограми $2\gamma_I(h)$ — залежності варіації значень яскравості пікселів зображення $I(\boldsymbol{s})$ від величини інтервалу (лага) h між ними:

$$2\gamma_I(h) = 2\big(C_I(0) - C_I(h)\big).$$

У більшості практичних застосувань ВА використовується наступна оцінка варіограми $2\gamma_I(h)$ [223]:

$$2\hat{\gamma}(h) = \frac{1}{|N_h|} \cdot \sum_{i,j \in N_h} \left(I(s_i) - I(s_j)\right)^2, \; N_h = \{(i,j): s_i - s_j = h\}, \qquad (6.18)$$

де N_h — множина всіх можливих позицій елементів сигналу, відстань між якими дорівнює h.

Використання оцінки варіограми $2\hat{\gamma}(h)$ (6.18) дозволяє суттєво спростити визначення кореляційних характеристик ЦЗ, оскільки не потребує використання апріорних даних щодо параметрів розподілу значень яскравості пікселів, які відповідають структурним $Y(\boldsymbol{s})$ і шумовим $\varepsilon(\boldsymbol{s})$ компонентам зображення.

ВА ЦЗ дозволяє визначити величину *nugget*-ефекту:

$$N_I = 2\hat{\gamma}(h)\,|_{h=1}; \qquad (6.19)$$

варіативність (*sill*) ряду, яка характеризує максимальне значення дисперсії яскравості пікселів зображення $I(\boldsymbol{s})$:

$$S_I = 2\hat{\gamma}(h)|_{h\to+\infty} \equiv \sigma_I^2 = \sigma_Y^2 + \sigma_\varepsilon^2, \; \boldsymbol{s} \in D_s; \qquad (6.20)$$

та інтервал кореляції (*range*) значень яскравості пікселів досліджуваного зображення [224]:

$$R_I = \max\left\{h: \left(1 - \frac{2\gamma_I(h)}{S_I}\right) \geq \varepsilon_R\right\}, \; \varepsilon_R \in \mathbb{R}_+. \qquad (6.21)$$

Значення R-параметра зазвичай визначається для випадку, коли кореляція $\rho(\boldsymbol{s}+h) = C_I(h)/C_I(0)$ є меншою за 10% ($\varepsilon_R = 0{,}1$) [223].

Використання S та R-параметрів варіограм $2\hat{\gamma}(h)$ дозволяє виявити зміни дисперсії та інтервалу кореляції значень яскравості пікселів ЦЗ, обумовлені вбудовування СД до ОПЦЗК згідно одноетапних, багатоетапних і комплексних методів, в окремих областях D_{part} $(D_{part} \subset D_s)$ матриць зображення $I(\boldsymbol{s})$.

Визначення N, S та R-параметрів кольорового зображення $\boldsymbol{I}_{x,y}$ розмірами $M \times N$ пікселів проводиться в декілька етапів [225]. На першому етапі проводиться формування послідовностей $g(\rho,\theta,x,y)$ яскравості пікселів окремих каналів кольору зображення $\boldsymbol{I}_{x,y}$ з використанням «скануючої» прямої $a_I(\rho,\theta,x,y)$:

$$g(\rho,\theta,x,y) = \boldsymbol{I}_{x,y} \times \delta_D\big(a_I(\rho,\theta,x,y)\big), \qquad (6.22)$$

$$a_I(\rho,\theta,x,y) = x \times \cos(\theta) + y \times \sin(\theta) - \rho,$$

де θ, ρ — відповідно, кут нахилу та відстань прямої $a_I(\rho,\theta,x,y)$ від полюса полярної системи координат; $\delta_D(\cdot)$ — функція Дірака (дельта-функція). Початок координат полярної системи встановлений на позиції

$(x, y) = (1,1)$ зображення $\boldsymbol{I}_{x,y}$. Використання «скануючої» прямої (6.22) дозволяє досліджувати кореляцію значень яскравості пікселів ЦЗ від просторового зсуву між ними та є аналогічним до відомого перетворення Радона [160].

Кількість елементів послідовності $g(\rho, \theta, x, y)$ (6.22) є максимальною при скануванні окремих каналів кольору зображення $\boldsymbol{I}_{x,y}$ по діагоналі ($\theta \in \left\{\frac{\pi}{4} + \frac{\pi}{2}n, n \in \mathbb{Z}\right\}$, $\rho = \frac{M \times N}{\sqrt{M^2+N^2}}$) та зменшується при зміні значень θ та ρ, що призводить до зниження точності оцінки параметрів N, S та R. Для забезпечення фіксованої точності визначення зазначених параметрів пропонується проводити розрахунок варіограм $2\hat{\gamma}(h)$ при обробці окремих рядків ($\theta \in \{\pi n, n \in \mathbb{Z}\}$, $\rho \in [1; M]$) або стовпчиків ($\theta \in \left\{\frac{\pi}{2} + \pi n, n \in \mathbb{Z}\right\}$, $\rho \in [1; N]$) каналів кольору зображення $\boldsymbol{I}_{x,y}$.

На другому етапі, згідно формули (6.18), при заміні $I(s_i)$ та $I(s_j)$ на отримані послідовності $g(\rho, \theta, x, y)$ (6.22), розраховуються варіограми $2\hat{\gamma}(h)$.

На третьому етапі проводиться апроксимація варіограм $2\hat{\gamma}(h)$ методом найменших квадратів з використанням наступних моделей [214, 223, 224]:

1) лінійна модель:

$$2\hat{\gamma}(h, \boldsymbol{v}) = v_1 + v_2 \times h, \tag{6.23}$$

де $\boldsymbol{v} = \{v_i : v_i > 0,\ i \in [1; 2]\}$ — вектор параметрів моделі;

2) сферична модель:

$$2\hat{\gamma} = \begin{cases} v_1, & h > v_3, \\ v_1 + v_2 \times \left[\frac{3}{2} \times \left(\frac{h}{v_3}\right) - \frac{1}{2} \times \left(\frac{h}{v_3}\right)^3\right], & 0 \le h \le v_3, \end{cases} \tag{6.24}$$

де $\boldsymbol{v} = \{v_i : v_i > 0,\ i \in [1; 3]\}$ — вектор параметрів моделі;

3) степенева модель:

$$2\hat{\gamma}(h, \boldsymbol{v}) = v_1 + v_2 \times h^{v_3}, \tag{6.25}$$

де $\boldsymbol{v} = \{v_i : v_i > 0,\ i \in [1; 2];\ v_3 \in [0; 2)\}$ — вектор параметрів моделі;

4) експоненційна модель:

$$2\hat{\gamma}(h, \boldsymbol{v}) = v_1 + v_2 \times \left[1 - e^{-\frac{h}{v_3}}\right], \tag{6.26}$$

де $\boldsymbol{v} = \{v_i : v_i > 0,\ i \in [1; 3]\}$ — вектор параметрів моделі;

5) гаусівська модель:

$$2\hat{\gamma}(h,\boldsymbol{v}) = v_1 + v_2 \times e^{\left(-\frac{(h-v_3)}{v_4^2}\right)}, \tag{6.27}$$

де $\boldsymbol{v} = \{v_i : v_i > 0,\ i \in [1;4]\}$ — вектор параметрів моделі;

6) хвильова (*wave*) модель:

$$2\hat{\gamma}(h,\boldsymbol{v}) = v_1 + v_2 \times \left[1 - \frac{\sin(h/v_3)}{(h/v_3)}\right], \tag{6.28}$$

де $\boldsymbol{v} = \{v_i : v_i > 0,\ i \in [1;3]\}$ — вектор параметрів моделі;

7) модель Матерна:

$$2\hat{\gamma}(h,\boldsymbol{v}) = C_I(0,\boldsymbol{v}) - C_I(h,\boldsymbol{v}), \tag{6.29}$$

$$C_I(h,\boldsymbol{v}) = v_1^2 \times [h=0]_I + v_2^2 \times \left[\frac{\{h/v_1\}^{v_2} \times K_{v_2}(h/v_1)}{2^{v_2-1} \times \Gamma(v_2)}\right],$$

$$\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt,\ \ x \in \mathbb{C}: \mathrm{Re}(x) > 0,$$

де $C_I(h,\boldsymbol{v})$ — функція коваріації Матерна; $K_{v_2}(\cdot)$ — модифікована функція Бесселя другого типу порядку v_2 [107]; $\Gamma(\cdot)$ — гамма-функція; $\boldsymbol{v} = \{v_i : v_i > 0,\ i \in [1;4]\}$ — вектор параметрів моделі.

На четвертому етапі для кожної апроксимації варіограми $2\hat{\gamma}^{appr}(h)$ згідно формул (6.23)–(6.29) проводиться розрахунок відповідної середньо-квадратичної помилки ε_{MSE}:

$$\varepsilon_{MSE} = \sqrt{\sum_{h=1}^{|N(h)|} \left(2\hat{\gamma}(h) - 2\hat{\gamma}^{appr}(h)\right)^2 / |N(h)|}. \tag{6.30}$$

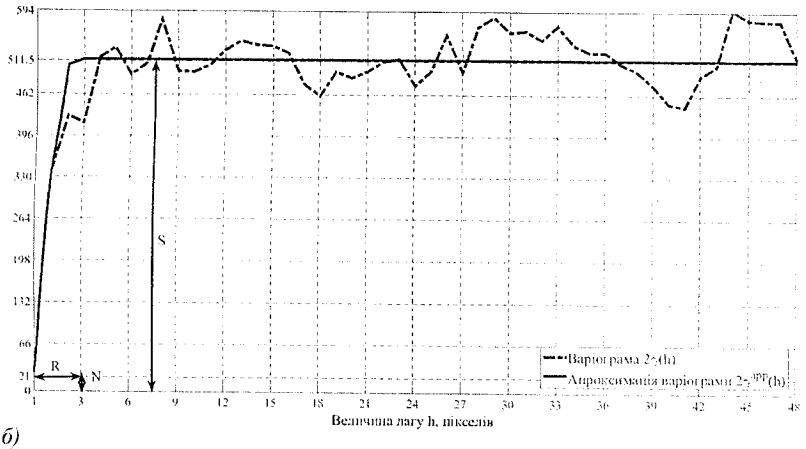
Використовуючи апроксимацію варіограми $2\hat{\gamma}^{appr}(h)$, для якої досягається мінімальне значення помилки ε_{MSE} (6.30), розраховуються значення *N*, *S* та *R*-параметрів для обраної послідовності $g(\rho,\theta,x,y)$ (6.22) згідно виразів (6.19)–(6.21).

На рис. 6.7 зображено профіль яскравості псевдовипадково обраного рядка каналу зеленого кольору тестового зображення (табл. 6.2), оцінка варіограми $2\hat{\gamma}(h)$ цього рядка та гаусівська апроксимація $2\hat{\gamma}^{appr}(h)$ варіограми $2\hat{\gamma}(h)$.

При апроксимації варіограми $2\hat{\gamma}(h)$ (рис. 6.7, *б*) було встановлено, що мінімальне значення помилки ε_{MSE} (6.30) досягається при використанні гаусівської моделі (6.27). Значення $2\hat{\gamma}^{appr}(h)$ при $h = 1$ характеризує величину *nugget*-ефекту (*N*-параметр); значення *S*-параметра визначається

рівнем насичення варіограми $2\hat{\gamma}(h)$ (рис. 6.7, *б*) і характеризує варіативність значень яскравості пікселів ЦЗ; значення *R*-параметра дорівнює довжині інтервалу кореляції значень яскравості пікселів на рівні $0.9 \times S$.

а)



б)

Рис. 6.7. Профіль яскравості псевдовипадково обраного рядка каналу зеленого кольору тестового зображення (*а*); оцінка варіограми $2\hat{\gamma}(h)$ цього рядка та гаусівська апроксимація $2\hat{\gamma}^{appr}(h)$ варіограми $2\hat{\gamma}(h)$ (*б*)

Значення *N*, *S* та *R*-параметрів, розраховані для гаусівської апроксимації $2\hat{\gamma}^{appr}(h)$ варіограми згідно формул (6.19)–(6.21), дорівнюють: $N = 21$, $S = 511.5$ та $R = 3$ (рис. 6.7, *б*).

Параметр *S*, що відповідає дисперсії значень яскравості елементів досліджуваного рядка (рис. 6.7, *а*), дозволяє оцінити середньоквадратичне відхилення значень яскравості пікселів $\sigma = \sqrt{S} \cong 22.62$, що узгоджується з результатом безпосереднього розрахунку:

$$\sigma = \sqrt{E\left[\left(\boldsymbol{I}_{256,x} - \boldsymbol{E}[\boldsymbol{I}_{256,x}]\right)^2\right]} \cong 22.38.$$

Значення ж N та R-параметрів ($N = 21, R = 3$) знаходяться в межах статистичних характеристик для зображень-контейнерів.

Беручи до уваги, що збільшення інтервалу значень лага $h \gg R$ ($h \in [1; M]$) практично не впливає на точність оцінки дисперсії значень яскравості пікселів (S-параметра), важливим є визначення максимального значення лага h_{max}, використання якого дозволить знизити час аналізу ЦЗ при забезпеченні фіксованої точності визначення S-параметра.

Для визначення максимального значення лага h_{max} було проведено дослідження змін кореляції $\rho(h)$ яскравості суміжних по горизонталі пікселів, при варіації відстані (лага) h між ними для рядків усіх каналів кольору 2500 тестових зображень (табл. 6.2). Нормована залежність $\rho(h)$, наведена на рис. 6.8.

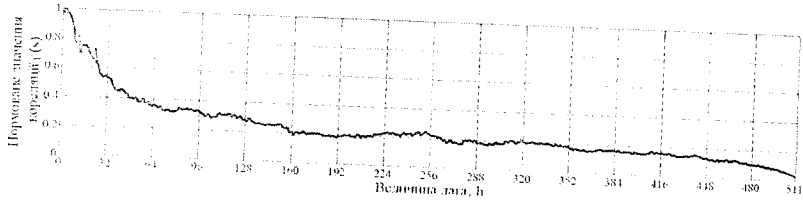


Рис. 6.8. Нормована кореляційна функція $\rho(h)$ залежності значень яскравості пікселів від відстані (лага) h між ними для рядків усіх каналів кольору 2500 тестових зображень

За результатами аналізу отриманих даних встановлено, що величина $\rho(h)$ зменшується в e разів вже для відстаней $h \in [96; 128]$ пікселів (рис. 6.8), а подальше зростання значення h не призводить до суттєвих змін значень нормованої кореляції $\rho(h)$. Отож, при дослідженні зображень максимальне значення лага h_{max} було обрано рівним $h_{max} = 128$.

Для визначення N, S та R-параметрів усіх каналів кольору ЦЗ $\boldsymbol{I}_{x,y}$ на основі ВА використано алгоритм [161], заснований на обчисленні вказаних параметрів для кожного рядка (стовпця) ЦЗ і подальшого розрахунку стандартних характеристик розподілу значень цих параметрів для кожного каналу кольору зображення $\boldsymbol{I}_{x,y}$:

1) центральних моментів вищих порядків — середнього значення, дисперсії, коефіцієнтів асиметрії та ексцесу;
2) оцінок середнього значення та дисперсії у випадку негаусівського розподілу значень N, S та R-параметрів — першого ($F_{0.25}$) та

третього ($F_{0.75}$) квартилів, медіани (квартиля $F_{0.50}$) та інтерквартильного розмаху;

3) параметрів варіативності – моди розподілу, розмаху варіації, коефіцієнтів осциляції та варіації, лінійного коефіцієнту варіації.

У табл. 6.6 наведено результати розрахунку *N*, *S* та *R*-параметрів для каналу синього кольору незаповненого ЦЗК та стеганограми, сформованої згідно методу Дея при слабкому заповненні ЦЗК стеганоданими ($\Delta_C = 10\%$) та мінімальній енергії повідомлень ($G = G_{min}$).

Таблиця 6.6

Характеристики розподілу значень *N*, *S* та *R*-параметрів для каналу синього кольору пустого ЦЗК та сформованої за методом Дея стеганограми, при $\Delta_C = 10\%$ та $G = G_{min}$

Характеристики розподілу значень	Параметри пустого ЦЗК:			Параметри стеганограми:		
	N	*S*	*R*	*N*	*S*	*R*
Середнє значення	136.762	5527.614	161.953	138.573	5622.602	162.0391
Дисперсія	13419.010	10326283.0	6515.683	13824.878	10676157.0	6523.802
Коефіцієнт асиметрії	1.019	0.245	–0.541	1.021	0.246	–0.542
Коефіцієнт ексцесу	–0.163	–0.963	–0.891	–0.160	–0.958	–0.893
1-й квартиль, $F_{0.25}$	65.053	2213.221	78.000	65.812	2252.697	78.000
Медіана (квартиль $F_{0.50}$)	91.267	5496.287	177.000	92.281	5597.450	177.000
3-й квартиль, $F_{0.75}$	199.413	8218.770	222.250	201.261	8357.867	222.250
Інтерквартильний розмах	134.360	6005.548	144.250	135.449	6105.170	144.250
Мода розподілу	93.710	1087.957	254.000	94.136	1107.904	254.000
Розмах варіації	431.378	13533.470	253.000	437.736	13768.646	253.000
Коефіцієнт осциляції	3.154	2.448	1.562	3.159	2.449	1.561
Коефіцієнт варіації	0.847	0.581	0.498	0.849	0.581	0.498
Лінійний коефіцієнт варіації	60.119370	2861.0962	77.000	60.888512	2903.614	77.000

Як видно з наведених у табл. 6.6 даних, приховання повідомлень у спектральній області ЦЗК з використанням двовимірного ДВП згідно методу Дея призводить до сильних змін усіх характеристик розподілу значень *N* та *S*-параметрів зображення-контейнера, що дозволяє використовувати для виявлення стеганограм прості порогові методи.

Для оцінки ймовірності виявлення стеганограм з даними, вбудованими до ОПЦЗК згідно одноетапних, багатоетапних і комплексних методів у [161] запропоновано варіограмний стеганодетектор SD_{VgA}, загальна кількість ознак якого (параметрів розподілу значень *N*, *S* та *R*-параметрів для кожного каналу кольору ЦЗ) склала $d_{VgA} = 13 \times 3 = 39$. Налаштування SD_{VgA} було проведено аналогічно до стеганодетекторів на основі статистичних моделей ЦЗК із застосуванням ансамблю класифікаторів. Згідно рекомендацій [191], в якості базових класифікаторів АК були використані ЛДФ. Визначення параметрів АК (кількості

базових класифікаторів і потужності множини ознак кожного БК), налаштування АК та ЛДФ проводилося згідно алгоритмів, наведених у додатках до [161].

Виявлення стеганограм, сформованих згідно одноетапних методів Дея та Агарвала, багатоетапних методів Джозефа та Хана, а також комплексних методів Елахіана та Ганджела, проводилося з використанням стеганодетектора SD_{VgA} на тестовому пакеті з 2500 цифрових зображень, обраних псевдовипадковим чином зі стандартного пакета *MIRFlickr-25000* (табл. 6.2). Сімейства залежностей значень метрики AUC від ступеня заповнення ЦЗК стеганоданими Δ_C та вагового коефіцієнта G для стеганодетектора SD_{VgA}, при вбудовуванні СД типу «Креслення», «Карта» й «Портрет» (табл. 6.2) до різних каналів кольору ЦЗК згідно одноетапних, багатоетапних і комплексних методів, мають однаковий характер і діапазоном значень метрики AUC відрізняються несуттєво [161]. У якості прикладу на рис. 6.9 наведені сімейства залежностей значень метрики AUC від ступеня заповнення ЦЗК стеганоданими Δ_C і вагового коефіцієнта G для стеганодетектора SD_{VgA}, при вбудовуванні СД типу «Портрет» до каналу синього кольору ЦЗК згідно одноетапних, багатоетапних і комплексних методів.

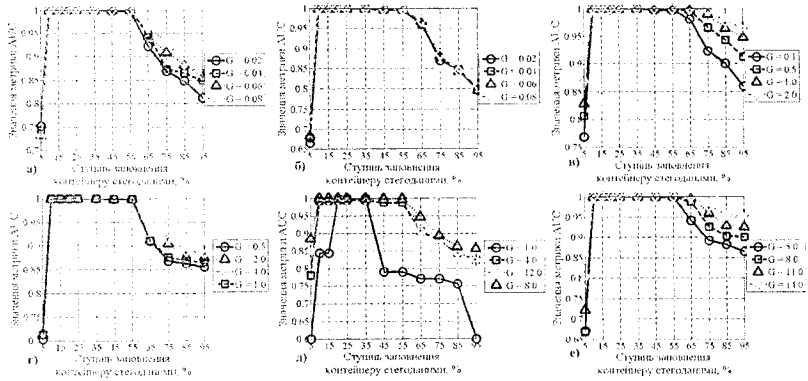


Рис. 6.9. Сімейства залежностей значень метрики AUC від ступеня заповнення ЦЗК СД і вагового параметра G, одержані для стеганодетектора SD_{VgA}, при вбудовуванні СД типу «Портрет» у канал синього кольору ЦЗК згідно: *(а)* – методу Дея; *(б)* – методу Агарвала; *(в)* – методу Джозефа; *(г)* – методу Хана; *(д)* – методу Елахіана; *(е)* – методу Ганджела

За результатами аналізу залежностей $AUC(\Delta_C, G)$, приклад яких наведено на рис. 6.9, встановлено, що застосування варіограмного аналізу

забезпечує високу ймовірність ($AUC \approx 0.99$) виявлення стеганограм з даними, вбудованими до спектральної області ЦЗК згідно методу Дея, навіть у випадку мінімальної енергії СД (рис. 6.9, *а*).

Результат порівняльного аналізу ймовірності виявлення (метрика AUC) стеганограм з даними, вбудованими до ОПЦЗК згідно одноетапних методів Дея та Агарвала, багатоетапних методів Джозефа та Хана, а також комплексних методів Елахіана та Ганджела, при використанні різних методів ПСА ЦЗ, зведено до табл. 6.7.

Таблиця 6.7

Метрика AUC та середній час обробки $\langle T \rangle$ ЦЗ при застосуванні SD_{CDF}, SD_{VgA} та УСД Авсібаша (гаусівська фільтрація) для виявлення СД типу «Портрет», вбудованих до каналу синього кольору ЦЗК згідно одноетапних, багатоетапних і комплексних методів, у випадку слабкого (AUC_{min}, $\Delta_C = 5\%$) при $G = G_{min}$ та сильного (AUC_{max}, $\Delta_C = 85\%$) при $G = G_{max}$ заповнення ЦЗК

		Статистична модель *CDF*	УСД Авсібаша	Варіограмний аналіз
Метод Дея	AUC_{max} / AUC_{min}	0.999 / 0.725	0.915 / 0.551	0.999 / 0.778
	$\langle T \rangle$, с	11.4	5.3	7.1
Метод Агарвала	AUC_{max} / AUC_{min}	0.993 / 0.724	0.646 / 0.514	0.999 / 0.665
	$\langle T \rangle$, с	11.8	5.8	7.4
Метод Джозефа	AUC_{max} / AUC_{min}	0.999 / 0.603	0.859 / 0.507	0.999 / 0.769
	$\langle T \rangle$, с	11.3	5.5	7.2
Метод Хана	AUC_{max} / AUC_{min}	0.999 / 0.997	0.999 / 0.997	0.999 / 0.703
	$\langle T \rangle$, с	11.2	5.2	7.1
Метод Елахіана	AUC_{max} / AUC_{min}	0.999 / 0.991	0.720 / 0.713	0.999 / 0.774
	$\langle T \rangle$, с	11.0	5.3	7.2
Метод Ганджела	AUC_{max} / AUC_{min}	0.999 / 0.998	0.943 / 0.671	0.999 / 0.668
	$\langle T \rangle$, с	11.1	5.2	7.1

Аналізуючи табл. 6.7, можна зробити наступні висновки:

- при слабкому заповненні ЦЗК стеганоданими ($\Delta_C \leq 10\%$) застосування варіограмного аналізу ЦЗ дозволяє підвищити ймовірність виявлення стеганограм (метрика AUC), сформованих згідно одноетапного методу Дея та двоетапного методу Джозефа, у порівнянні з відповідними результатами для СД на основі статистичної моделі *CDF* та УСД Авсібаша;
- у випадку сильного заповнення ЦЗК стеганоданими ($\Delta_C = 85\%$) використання варіограмного аналізу дає можливість досягти значень метрики AUC, співставних з відповідними результатами для стеганo-

детектора SD_{CDF}, проте з використанням суттєво меншої кількості ознак — $d_{vgA} = 39$ проти $d_{CDF} = 1234$;

- також вагомою перевагою використання ВА є скорочення середнього часу обробки окремих каналів кольору ЦЗ на 36% у порівнянні з випадком використання стеганодетектора на основі статистичної моделі CDF — $\langle T_{VgA} \rangle \approx 7.2$ с і $\langle T_{CDF} \rangle \approx 11.3$ с — унаслідок використання простору ознак стеганодетектора суттєво меншої розмірності.

Формування стеганограм із застосуванням сингулярного розкладу матриць яскравості пікселів зображення-контейнера згідно методів Агарвала та Джозефа (табл. 6.1) призводить до мінімальних змін кореляційних характеристик ЦЗК. При цьому для виявлення слабких змін кореляційних характеристик шумів зображення-контейнера, зумовлених вбудовуванням СД згідно зазначених СМ, перспективним бачиться використання спеціальних методів структурного аналізу, зокрема — флуктуаційного.

6.6.2.2. *Флуктуаційний аналіз цифрових зображень*

Для дослідження кореляції між елементами дискретного сигналу широко використовується показник Герста (*Hurst exponent*) [221, 226], значення H якого для сигналів змінюється в інтервалі $H \in [0; 1]$, де випадку $H = 1/2$ відповідає відсутність кореляції між елементами послідовності. У літературі прийнято, що належність показника Герста інтервалові $H \in (1/2; 1]$ відповідає позитивній кореляції значень суміжних елементів досліджуваного сигналу, а $H \in [0; 1/2)$ — негативній кореляції [221].

Один з найбільш відомих методів визначення показника Герста — т. зв. *Rescaled Range (R/S)* аналіз [226]. Згідно *R/S* аналізу оцінка значення показника Герста H проводиться з використанням середньоквадратичного відхилення S_X та розмаху R_X варіації значень заданої послідовності X довжиною K елементів:

$$H \approx \frac{lg\left[E\left[\frac{R_X}{S_X}\right]\right]}{lg[K]}, \tag{6.31}$$

$$R_X(t) = \max\left[\sum_{i=1}^{t}(x_i - E_K[X])\right] - \min\left[\sum_{i=1}^{t}(x_i - E_K[X])\right], \quad t \in [1; K],$$

$$S_X(t) = \sqrt{\frac{\sum_{i=1}^{t}(x_i - E_K[X])}{t}}, \quad t \in [1; K],$$

де $E_m[X] = \frac{1}{m}\sum_{j=1}^{m} x_j$, $m \in [1; K]$ — середнє значення, розраховане для перших m елементів заданої послідовності.

При визначенні показника Герста згідно формули (6.31) робиться припущення, що при розбитті досліджуваного сигналу на окремі інтервали (фрагменти) значення показника залежить лише від ширини фрагмента w. Проте для більшості реальних сигналів, у тому числі й цифрових зображень, зазначене припущення не виконується внаслідок наявності як «короткотривалої» (при $w \leq 10$), так і «довготривалої» (при $w \gg 10$) кореляції між значеннями елементів досліджуваного сигналу. Оцінка величини зазначених кореляцій потребує переходу від показника Герста H до спектра узагальнених експонент Герста (СУЕГ) h_q [222, 227]:

$$h_q \approx \frac{lg\left[E\left[\left(X(t+\tau) - X(t)\right)^q\right]\right]}{lg[\tau]},$$

де τ — значення просторового зсуву (лага) між пікселями досліджуваного зображення; $q(q \in \mathbb{R})$ — масштабуюча експонента. Варіація значення параметра q дає можливість оцінювати значення узагальненої експоненти h_q для флуктуацій значення яскравості пікселів ЦЗ як малої $(q < 0)$, так і великої $(q > 0)$ амплітуди.

Для розрахунку СУЕГ використовуються модифікації відомих методів визначення показника Герста [221, 226, 228–231], зокрема флуктуаційний аналіз з компенсацією тренду (*Detrended Fluctuation Analysis, DFA*) й аналіз локальних екстремумів дискретного вейвлет перетворення досліджуваного сигналу (*Wavelet Transform Modulus Maxima, WTMM*) [221, 232, 254]. Особливістю цих методів є представлення шумів у сигналі X як мультифракталу — суперпозиції декількох монофрактальних компонент, статистичні характеристики яких є однаковими на різних масштабах дослідження сигналу X [233]. Це дозволяє використовувати зазначені методи для оцінки як кореляційних, так і фрактальних характеристик шумових компонент досліджуваних сигналів, що представляє особливий інтерес при проведенні ПСА.

Аналіз *WTMM* є одним з найбільш застосовних при визначенні як показника Герста H, так і СУЕГ h_q, і заснований на використанні узагальненої статистичної суми $Z(q,s)$ для дослідження характеристик розподілу значень локальних максимумів $\left|W_X^{\psi}(s,a)\right|$ вейвлет перетворення сигналу $\boldsymbol{X}(n)$ на масштабі s та зсуві a базисного вейвлета $\psi(n)$:

$$Z(q,s) = \sum_{j=1}^{j_{max}} \left|W_X^{\psi}(s,a)\right|^q, \qquad (6.32)$$

де j_{max} — заданий максимальний рівень декомпозиції сигналу X.

Вагомим обмеженням застосування аналізу *WTMM* при проведенні ПСА цифрових зображень є неможливість визначення кореляційних характеристик шумових компонент ЦЗ, що обумовлено використанням лише локальних максимумів $W_X^{\psi}(s,a)$ коефіцієнтів ДВП сигналу при розрахунку узагальненої статистичної суми $Z(q,s)$ (6.32). Унаслідок цього коефіцієнти вейвлет перетворення сигналу X з малою амплітудою, на рівні яких проводиться приховання повідомлень, до уваги не беруться.

Для дослідження кореляційних і фрактальних характеристик сигналу X — профілю яскравості рядків (стовпчиків) ЦЗ — пропонується використати модифікований метод флуктуаційного аналізу [222, 234] — мультифрактальний флуктуаційний аналіз (МФФА). МФФА одновимірного сигналу $\boldsymbol{X}_{1\times L_X}$ довжиною L_X відліків проводиться в декілька етапів [222]. На першому етапі розраховується послідовність кумулятивних сум $\boldsymbol{X}_{cml}(n)$ значень елементів сигналу $\boldsymbol{X}$:

$$\boldsymbol{X}_{cml}(j) = \sum_{i=1}^{j} \boldsymbol{X}(i), \quad j \in [1; L_X]. \tag{6.33}$$

Отримана послідовність $\boldsymbol{X}_{cml}(n)$ розбивається на інтервали, що не перетинаються, довжиною w елементів — від початку сигналу до його кінця, та в зворотному напрямку. Це дозволяє підвищити потужність множини комбінацій елементів досліджуваного сигналу $\boldsymbol{X}(n)$ та, відповідно, точність визначення узагальнених експонент Герста (УЕГ) h_q у випадку аналізу відносно коротких сигналів ($L_X < 100$).

На другому етапі МФФА проводиться розрахунок дисперсії значень $F^2(w,v)$ для v-го інтервалу розбиття послідовності $\boldsymbol{X}_{cml}(n)$:

$$F^2(w,v) = \begin{cases} \dfrac{\sum_{i=1}^{w}\left\{\boldsymbol{X}_{cml}[(v-1)\times w + i] - \boldsymbol{y}_v^{appr}(i)\right\}^2}{w}, & v \in [1; N_w]; \\ \dfrac{\sum_{i=1}^{w}\left\{\boldsymbol{X}_{cml}[L_X - (v-N_w)\times w + i] - \boldsymbol{y}_v^{appr}(i)\right\}^2}{w}, & v \in [N_w+1; 2N_w], \end{cases} \tag{6.34}$$

де $N_w = \lfloor L_X/w \rfloor$ — кількість інтервалів розбиття сигналу $\boldsymbol{X}_{cml}(n)$, отриманих при розбитті послідовності $\boldsymbol{X}_{cml}(n)$ в одному напрямку (від початку сигналу до кінця або в зворотному напрямку); $\boldsymbol{y}_v^{n_{appr}}(n)$ — поліном степеня n_{appr}, що використовується для компенсації детермінованих компонент (тренду) в заданому інтервалі розбиття.

Віднесення зображень до класу пустих або заповнених ЦЗК потребує дослідження нелінійних залежностей між значеннями яскравості суміжних пікселів у їхньому складі. Апроксимація детермінованих компонент послідовності $\boldsymbol{X}_{cml}(n)$ в окремих інтервалах розбиття з використанням поліномів високого степеня ($n_{appr} \gg 1$) призводить до компенсації цих

нелінійних залежностей [235], що знижує точність виявлення стеганограм при проведенні ПСА. Для мінімізації змін нелінійних залежностей між значеннями яскравості суміжних пікселів досліджуваного зображення при компенсації детермінованих компонент в окремих інтервалах розбиття послідовності $\boldsymbol{X}_{cml}(n)$ нами використовувалася лінійна апроксимація ($n_{appr} = 1$).

На третьому етапі розраховується функція флуктуації $F_q(w)$, шляхом усереднення дисперсії $F^2(w, v)$ (6.34) по всіх інтервалах розбиття послідовності $\boldsymbol{X}_{cml}(n)$:

$$F_q(w) = \begin{cases} \left\{ \frac{1}{2N_w} \cdot \sum_{v=1}^{2N_w} [F^2(w,v)]^{\frac{q}{2}} \right\}^{\frac{1}{q}}, & q \in \mathbb{R} \backslash \{0\}; \\ \exp\left\{ \frac{1}{4N_w} \cdot \sum_{v=1}^{2N_w} \ln[F^2(w,v)] \right\}, & q = 0. \end{cases} \tag{6.35}$$

Розрахунок функції флуктуації $F_q(w)$ повторюється при варіації значень ширини w інтервалів розбиття послідовності $\boldsymbol{X}_{cml}(n)$ не менше трьох разів.

На четвертому етапі здійснюється побудова залежності усередненої дисперсії $F_q(w)$ (6.35) від інтервалів w блоків розбиття послідовності $\boldsymbol{X}_{cml}(n)$ у логарифмічних координатах — $lg[F_q(w)] = f(lg[w])$. Значення узагальненої експоненти Герста h_q при цьому дорівнює тангенсу кута нахилу прямої, що апроксимує залежність $F_q(w)$.

На п'ятому етапі розраховується мультифрактальний спектр (МФС) $f_q(\alpha_q)$, шляхом застосування до масштабуючих показників Реньї (*Rényi dimensions*) $\tau(q) = q \times h_q - 1$ прямого перетворення Лежандра (*Legendre transformation*) [220, 236]:

$$\alpha_q = \frac{d\tau(q)}{dq}; \tag{6.36}$$

$$f_q(\alpha_q) = q \times \frac{d\tau(q)}{dq} - \tau(q). \tag{6.37}$$

Дискретні значення $f_q(\alpha_q)$ відповідають фрактальній розмірності — т. зв. розмірності Гаусдорфа-Безіковича (РГБ) (*Hausdorff-Besikovytch dimension*) — підмножини відліків досліджуваного сигналу $\boldsymbol{X}(n)$, показник Гьольдера (*Hölder exponent*) якого дорівнює α_q [221] і варіюється від $\alpha_q = \alpha_q^{min}$ (що відповідає компонентам ЦЗ з найменшими

флуктуаціями значень яскравості пікселів) до $\alpha_q = \alpha_q^{max}$ (що відповідає найбільш «нерегулярним» компонентам ЦЗ).

На рис. 6.10 наведено СУЕГ h_q (рис. 6.10, *а*) та відповідний йому МФС $f_q(\alpha_q)$ (рис. 6.10, *б*) для псевдовипадково обраного рядка $\boldsymbol{X}(n)$ (рис. 6.7, *а*) каналу зеленого кольору тестового ЦЗ.

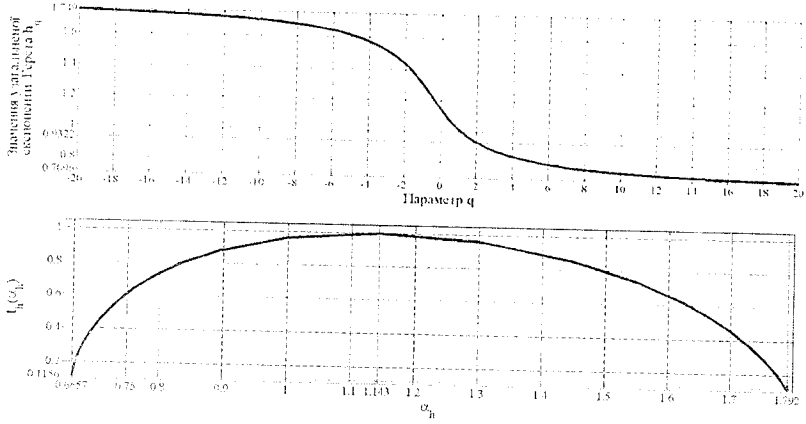


Рис. 6.10. Спектр узагальнених експонент Герста h_q *(а)* та відповідний мультифрактальний спектр $f_q(\alpha_q)$ *(б)* псевдовипадково обраного рядка $\boldsymbol{X}(n)$ каналу зеленого кольору тестового зображення $\boldsymbol{I}_{x,y}$

Взаємооднозначний зв'язок через пряме й зворотне перетворення Лежандра між СУЕГ h_q та МФС $f_q(\alpha_q)$ [220, 236] дозволяє визначити граничні значення показника Гьольдера α_q, використовуючи граничні значення УЕГ $h_{-\infty}$ та $h_{+\infty}$:

$$\left.\frac{d\tau(q)}{dq}\right|_{q\to(+\infty)} = h_{+\infty} = \alpha_q^{min};$$

$$\left.\frac{d\tau(q)}{dq}\right|_{q\to(-\infty)} = h_{-\infty} = \alpha_q^{max}.$$

За результатами проведених розрахунків нами визначені як граничні значення УЕГ $h_{-\infty}$ та $h_{+\infty}$, так і значення показника Гьольдера α_q^{min} та α_q^{max} (рис. 6.10):

$$h_{-\infty} \approx 1.749\,,\ h_{+\infty} \approx 0.7096;$$
$$\alpha_q^{max} \approx 1.792,\ \alpha_q^{min} \approx 0.6657.$$

Незначні відмінності між розрахованими значеннями $h_{-\infty}$ $(h_{+\infty})$ та α_q^{max} $\left(\alpha_q^{min}\right)$ пояснюються використанням скінченного діапазону значень параметра q.

Перевагою застосування МФФА при проведенні ПСА ЦЗ є висока точність оцінки ступеня «нерегулярності» й діапазону змін фрактальної розмірності шумових компонент рядків (стовпчиків) $\boldsymbol{X}(n)$ зображення при використанні ширини Δ_{α_q}:

$$\Delta_{\alpha_q} = \alpha_q^{max} - \alpha_q^{min}, \tag{6.38}$$

й розмаху Δ_{f_q}:

$$\Delta_{f_q} = f_q^{max}(\alpha_q) - f_q^{min}(\alpha_q), \tag{6.39}$$

мультифрактального спектра $f_q(\alpha_q)$ (рис. 6.10,*б*).

Теоретично, при проведенні МФФА зображень значення масштабуючих параметрів q_{min} і q_{max} мають прямувати, відповідно, до $q_{min} \to -\infty$ і $q_{max} \to +\infty$. Проте, практична реалізація зазначених вимог є неможливою через обмеження кількості бітів, використовуваних для представлення дійсних чисел при проведенні обчислень на комп'ютері.

За результатами оцінки точності визначення СУЕГ і МФС при проведенні МФФА зображень з використання тестового пакета ЦЗ (табл. 6.2) нами було встановлено, що граничні значення $h_{-\infty}$ та $h_{+\infty}$ УЕГ при збільшенні абсолютного значення параметра q $(|q| > 20)$ змінювалися менше, ніж на 0,01. Внаслідок цього при виявленні стеганограм з даними, вбудованими до ОПЦЗК згідно одноетапних, багатоетапних і комплексних методів, значення масштабуючого параметра q змінювалися в діапазоні $q \in [-20; 20]$, з кроком $\Delta_q = 0{,}5$.

Для зменшення тривалості обробки ЦЗ з використанням МФФА при забезпеченні фіксованої точності визначення СУЕГ також було проведено дослідження залежності усередненої похибки δ визначення значень УЕГ h_q від кроку вибору рядків Δ_{row} усіх каналів кольору 2500 зображень з тестового пакета (табл. 6.2) [237]:

$$\delta(k) = \frac{1}{N}\sum_{i=1}^{N}\left|S_1\left(i \times \Delta_q\right) - S_k\left(i \times \Delta_q\right)\right|,$$

де S_1, S_k — спектри узагальнених експонент Герста, усередненні по вибраних рядках ЦЗ, при, відповідно, $\Delta_{row} = 1$ і $\Delta_{row} = k$, $k \in [1; M]$; $M = 512$ — кількість рядків тестового зображення; $N = \frac{(q_{max} - q_{min})}{\Delta_q}$ — кількість порівнюваних складових спектра УЕГ.

Залежність похибки δ від кроку Δ_{row} вибору рядків тестових зображень наведена на рис. 6.11.

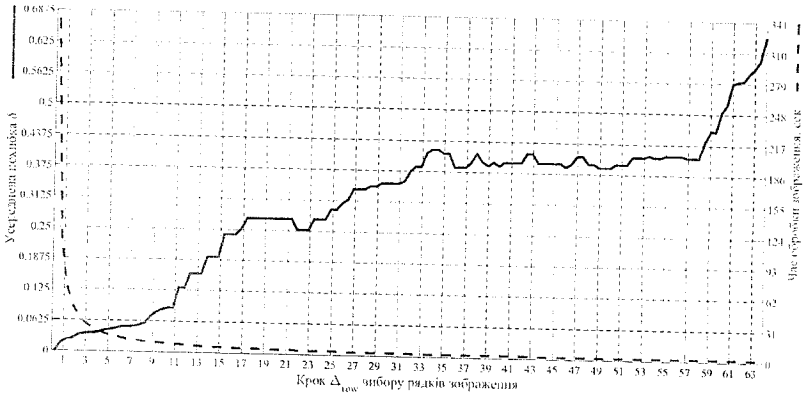


Рис. 6.11. Залежність усередненої похибки δ та часу обробки каналів кольору зображення, розміром 512×512 пікселів, від кроку вибору рядків при використанні МФФА: величина похибки (ліва вісь ординат, суцільна лінія); середній час обробки зображення (права вісь ординат, штрихова лінія)

За результатами аналізу отриманих даних (рис. 6.11) значення кроку Δ_{row} вибору рядків ЦЗ було обрано рівним $\Delta_{row} = 4$, що дало можливість забезпечити високу точність визначення СУЕГ ($\delta \approx 0.042$) при скороченні часу обробки на 93% — від 335 с при $\Delta_{row} = 64$ до 24,47 с при $\Delta_{row} = 4$.

Для визначення кореляційних і фрактальних характеристик шумових компонент цифрових зображень, на основі МФФА розроблений алгоритм обробки рядків (стовпчиків) ЦЗ [161]. Основними кроками зазначеного алгоритму є:

1) вибір рядків (стовпчиків) окремих каналів кольору досліджуваного ЦЗ із кроком Δ_{row} (Δ_{column});
2) розрахунок спектрів УЕГ h_q та мультифрактальних спектрів $f_q(\alpha_q)$ рядків (стовпчиків) ЦЗ з використанням одновимірного МФФА;
3) розрахунок основних характеристик спектрів h_q та $f_q(\alpha_q)$:
   a) показника Герста $H \equiv h_2$;
   b) граничних узагальнених експонент Герста $h_{-\infty}(h_{+\infty})$;
   c) середніх $(E[\acute{h}_q])$ та максимальних $(\max[\acute{h}_q])$ значень похідних СУЕГ при зміні параметра q в діапазонах $q \in [q_{min}; 0]$ та $q \in [0; q_{max}]$;

d) максимального ($f_q^{max}(\alpha_q)$, α_q^{max}) та мінімального ($f_q^{min}(\alpha_q)$, α_q^{min}) значень розмірностей Гаусдорфа-Безіковича та показника Гьольдера монофрактальних компонент шумів ЦЗ;

e) середніх ($E[\acute{f}_q(\alpha_q)]$) та максимальних ($\max[\acute{f}_q(\alpha_q)]$) значень похідних МФС;

4) розрахунок стандартних характеристик розподілу значень параметрів СУЕГ і МФС по рядках (стовпчиках) досліджуваного зображення:

a) центральних моментів вищих порядків — середнього значення, дисперсії, коефіцієнтів асиметрії та ексцесу;

b) оцінок середнього значення та дисперсії у випадку негаусівського розподілу значень *N*, *S* та *R*-параметрів — першого ($F_{0.25}$) і третього ($F_{0.75}$) квартилів, медіани (квартиля $F_{0.50}$), а також інтерквартильного розмаху;

c) параметрів варіативності — моди розподілу, розмаху варіації, коефіцієнтів осциляції та варіації, лінійного коефіцієнту варіації.

Для оцінки ймовірності виявлення стеганограм з даними, вбудованими до ОПЦЗК згідно одноетапних, багатоетапних і комплексних методів, на основі цього алгоритму був розроблений стеганодетектор SD_{MFDFA}. Загальна кількість ознак розробленого стеганодетектора — параметрів розподілу значень параметрів СУЕГ h_q та МФС $f_q(\alpha_q)$ кожного каналу кольору ЦЗ — склала $d_{MFDFA} = 14 \times 13 = 182$. Налаштування SD_{MFDFA} є аналогічним до налаштування стеганодетектора SD_{VgA} із застосуванням ансамблю класифікаторів (див. підп. 6.6.2.1). Визначення параметрів АК — кількості базових класифікаторів (лінійних дискримінантів Фішера) і потужності множини ознак кожного БК, — а також налаштування АК та ЛДФ проводилося згідно алгоритмів, наведених у Додатках [161].

Виявлення стеганограм, сформованих згідно одноетапних методів Дея та Агарвала, багатоетапних методів Джозефа та Хана, а також комплексних методів Елахіана та Ганджела, проводилося з використанням SD_{MFDFA} на тестовому пакеті з 2500 цифрових зображень (табл. 6.2). Сімейства залежностей значень метрики *AUC* від ступеня заповнення ЦЗК стеганоданими Δ_C і вагового коефіцієнта *G* для стеганодетектора SD_{MFDFA}, при вбудовуванні СД типу «Креслення», «Карта» й «Портрет» (табл. 6.2) до різних каналів кольору ЦЗК згідно одноетапних, багатоетапних і комплексних методів, мають однаковий характер та несуттєво різняться діапазоном значень метрики *AUC* [161]. В якості прикладу, на рис. 6.12 наведено приклади залежностей, отриманих при вбудовуванні СД типу «Портрет» до каналу синього кольору ЦЗК згідно одноетапних, багатоетапних та комплексних методів.

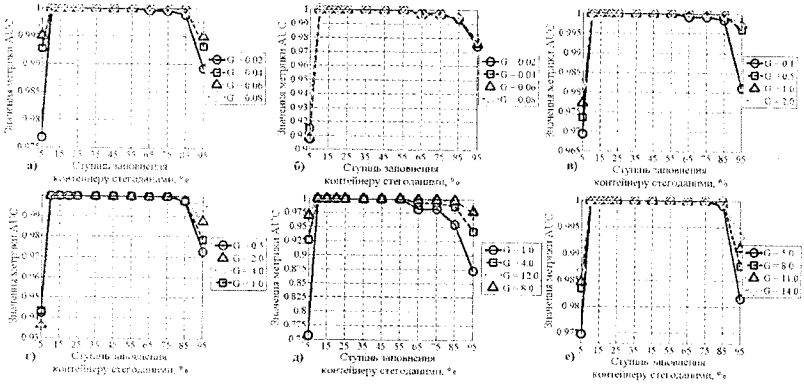


Рис. 6.12. Сімейства залежностей значень метрики AUC від ступеня заповнення ЦЗК стеганоданими і вагового параметра G, одержані для стеганодетектора SD_{MFDFA}, при вбудовуванні СД типу «Портрет» до каналу синього кольору ЦЗК згідно: *(а)* – методу Дея; *(б)* – методу Агарвала; *(в)* – методу Джозефа; *(г)* – методу Хана; *(д)* – методу Елахіана; *(е)* – методу Ганджела

Перевагою застосування МФФА для виявлення стеганограм з даними, вбудованими до ОПЦЗК, у порівнянні з ВА є розширення діапазону значень ступеня заповнення ЦЗК стеганоданими, чим забезпечується висока точність виявлення прихованих повідомлень: від $\Delta_C^{VgA} \in [10; 55]$ (рис. 6.9) до $\Delta_C^{MFDFA} \in [10; 85]$ (рис. 6.12).

Нами було проведено порівняльний аналіз імовірності виявлення (метрика AUC) стеганограм з даними, вбудованими до ОПЦЗК згідно одноетапних методів Дея та Агарвала, багатоетапних методів Джозефа та Хана, а також комплексних методів Елахіана та Ганджела, при використанні різних методів ПСА ЦЗ. Значення метрики AUC та середня тривалість обробки T стеганограм при застосуванні стеганодетекторів SD_{CDF} і SD_{MFDFA}, а також УСД Авсібаша (за гаусівської фільтрації ЦЗ) для виявлення СД типу «Портрет», вбудованих до каналу синього кольору зображення-контейнеру згідно одноетапних, багатоетапних і комплексних методів, у випадках слабкого (AUC_{min}, $\Delta_C = 5\%$) при $G = G_{min}$ та сильного (AUC_{max}, $\Delta_C = 85\%$) при $G = G_{max}$ заповнення ЦЗК стеганоданими наведені у табл. 6.8.

У порівнянні з випадком використання ВА (табл. 6.7), застосування МФФА дало можливість підвищити значення метрики AUC у випадку слабкого заповнення ($\Delta_C = 5\%$) й мінімальних значень вагового параметра G для одноетапних методів Дея ($\Delta AUC \approx 0.199$) та Агарвала ($\Delta AUC \approx 0.243$), а також двоетапного методу Джозефа ($\Delta AUC \approx 0.2$) приховання повідомлень до ОПЦЗК (табл. 6.8).

Таблиця 6.8

Значення AUC і середній час обробки $\langle T \rangle$ ЦЗ при застосуванні SD_{CDF}, SD_{MFDFA} та УСД Авсібаша (гаусівська фільтрація) для виявлення СД типу «Портрет», вбудованих до каналу синього кольору ЦЗК згідно одностанних, багатостанних і комплексних методів, у випадку слабкого (AUC_{min}, $\Delta_C = 5\%$) при $G = G_{min}$ та сильного (AUC_{max}, $\Delta_C = 85\%$) при $G = G_{max}$ заповнення ЦЗК

		Статистична модель *CDF*	УСД Авсібаша	МФФА
Метод Дея	AUC_{max} / AUC_{min}	0.999 / 0.725	0.915 / 0.551	0.999 / 0.977
	$\langle T \rangle$, с	11.4	5.3	9.7
Метод Агарвала	AUC_{max} / AUC_{min}	0.993 / 0.724	0.646 / 0.514	0.999 / 0.908
	$\langle T \rangle$, с	11.8	5.8	10.1
Метод Джозефа	AUC_{max} / AUC_{min}	0.999 / 0.603	0.859 / 0.507	0.999 / 0.969
	$\langle T \rangle$, с	11.3	5.5	9.8
Метод Хана	AUC_{max} / AUC_{min}	0.999 / 0.997	0.999 / 0.997	0.999 / 0.936
	$\langle T \rangle$, с	11.2	5.2	9.5
Метод Елахіана	AUC_{max} / AUC_{min}	0.999 / 0.991	0.720 / 0.713	0.999 / 0.756
	$\langle T \rangle$, с	11.0	5.3	9.6
Метод Ґанджела	AUC_{max} / AUC_{min}	0.999 / 0.998	0.943 / 0.671	0.999 / 0.975
	$\langle T \rangle$, с	11.1	5.2	9.5

Підвищення розмірності простору ознак — кореляційних і фрактальних характеристик шумових компонент ЦЗ — для стеганодетектора SD_{MFDFA} ($d_{MFDFA} = 182$, $\langle T_{MFDFA} \rangle \approx 9.7$ с) призвело до відповідного зростання на 35% середнього часу обробки зображення у порівнянні з стеганодетектором SD_{VgA} ($d_{VgA} = 39$, $\langle T_{VgA} \rangle \approx 7.2$ с), проте поступається середній тривалості обробки зображення з використанням статистичного стеганоаналізу ($T_{CDF} \approx 11.3$ с, табл. 6.8).

Використання декількох складових зображення-контейнера, що відповідають різним сингулярним числам (6.6), при формуванні стеганограм згідно методу Агарвала дозволяє зменшити ймовірність виявлення прихованих повідомлень з використанням методів статистичного стеганоаналізу та УСД Авсібаша, а також варіограмного (табл. 6.7) і мультифрактального флуктуаційного (табл. 6.8) аналізу. Виявлення змін статистичних характеристик окремих компонент ЦЗ, обумовлених вбудовуванням СД потребує використання спеціальних методів структурного аналізу ЦЗ, зокрема — мультифрактального.

6.6.2.3. Мультифрактальний аналіз стеганограм

У [233] Б. Мандельбротом було вперше доведено, що більшості процесів у реальному світі, приміром, зміні вартості цінних паперів на біржі, притаманна властивість самоподібності (фрактальності) — зберігання статистичних характеристик на різних масштабах дослідження реалізацій даних процесів. Сигнали, статистичні характеристики яких є однаковими на будь-якому масштабі дослідження, прийнято відносити до класу монофракталів [220]. У протилежному випадку сигнали є мультифрактальними та потребують використання декількох монофракталів для представлення досліджуваних сигналів із фіксованою точністю.

Роботи Б. Мандельброта поклали початок напрямку мультифрактального аналізу складних сигналів, який сьогодні широко використовується в різних галузях науки та техніки, зокрема комп'ютерній графіці [238], стисненні ЦЗ [238, 239], обробці сигналів і зображень [240–243], текстурному аналізі [244], аналізі стохастичних сигналів [245–247], дослідженні життєвих показників людини [248–250]. Зважаючи на високу «чутливість» результатів мультифрактального аналізу до будь-яких змін досліджуваних сигналів, перспективним бачиться й використання МФА для проведення пасивного стеганоаналізу ЦЗ.

Визначення мультифрактальних характеристик зображення $\boldsymbol{I}_{x,y}$ в деякій обмеженій області $\Im(\Im \subset \mathbb{R}^2)$ проводитимемо аналогічно до [236]. Розглянемо покриття області $\Im$ блоками $C_w(C_w \subset \mathbb{R}^2)$ зі сторонами $w(w \in \mathbb{N})$, що не перетинаються:

$$\Im \subset \bigcup_{i=1}^{N_{C_w}^{\Im}} C_w^i,$$

де $N_{C_w}^{\Im}$ — кількість блоків C_w, що покривають собою область $\Im$.

Для визначення ймовірності $p_i(w)$ того, що випадковим чином вибране значення яскравості пікселя ЦЗ знаходиться в блоці C_w^i, скористаємося узагальненою статистикою [236]:

$$p_i(w) = e^{-b_i(w)},\ i \in \left[1; N_{C_w}^{\Im}\right],$$

де $b_i(w)$ — середня яскравість пікселів зображення у блоці C_w^i, $i \in \left[1; N_{C_w}^{\Im}\right]$ з центром $\boldsymbol{x}(\boldsymbol{x} \in \mathbb{R}^2)$. Тоді «супроводжуючий» розподіл імовірностей $P_i(w,q)$ порядку $q(q \in \mathbb{R})$ можна представити в наступному вигляді [236]:

$$P_i(w,q) = \frac{p_i^q(w)}{\sum_{j=1}^{N_{C_w}^{\Im}} p_j^q(w)}, \quad i \in \left[1; N_{C_w}^{\Im}\right]. \tag{6.40}$$

«Супроводжуючий» розподіл (*escort distribution*) $P_i(w,q)$ використовується для вивчення структури розподілу значень яскравості пікселів зображення $\boldsymbol{I}_{x,y}$ по окремих комірках C_w^i, $i \in \left[1; N_{C_w}^{\Im}\right]$ шляхом варіації значень масштабуючого параметра $q (q \in \mathbb{R})$ [220]. У випадку $q \to +\infty$ основний внесок у розподіл $P_i(w,q)$ (6.40) вносять ті комірки, що мають велике значення середньої яскравості пікселів, тоді як вплив комірок з малим значенням середньої яскравості пікселів зростає при $q \to -\infty$.

Шляхом послідовних еквівалентних перетворень виразу (6.40) «супроводжуючий» розподіл $P_i(w,q)$ можна представити в формі канонічного розподілу Гіббза [236]:

$$Z(w,q) = \sum_{i=1}^{N_{C_w}^{\Im}} e^{-q \times b_i(w)} = \sum_{i=1}^{N_{C_w}^{\Im}} p_i^q(w), \tag{6.41}$$

$$\Psi(w,q) = -\ln[Z(w,q)],$$

$$P_i(w,q) = e^{\Psi(w,q) - q \times b_i(w)}, \quad i \in \left[1; N_{C_w}^{\Im}\right].$$

де $Z(w,q)$ — узагальнена статистична сума; $\Psi(w,q)$ — вільна енергія Гельмгольца.

Тоді інформація (ентропія) Реньї I_q порядку q є рівною [236]:

$$I_q(\Im, w) = \frac{\ln\left[\sum_{i=1}^{N_{C_w}^{\Im}} p_i^q(w)\right]}{q-1} = \left(-\frac{1}{q-1}\right) \times \Psi(w,q). \tag{6.42}$$

Ентропія Реньї I_q є узагальненням класичної ентропії Шеннона [65, 115, 220] і дозволяє визначати ступінь «випадковості» (стохастичності) розподілу значень яскравості пікселів зображення при варіації масштабуючого параметра q, що відповідає дослідженню компонент ЦЗ з різним рівнем середньої яскравості пікселів у блоках розбиття C_w^i, $i \in \left[1; N_{C_w}^{\Im}\right]$. Проте, обмеженням використання ентропії Реньї (6.42) при проведенні ПСА ЦЗ є залежність значень $I_q(\Im, w)$ від розмірів w блоків розбиття C_w^i, $i \in \left[1; N_{C_w}^{\Im}\right]$. Подолання даного обмеження потребує переходу від ентропії Реньї $I_q(\Im, w)$ до спектра узагальнених фрактальних розмірностей (СУФР) D_q [220, 236]:

$$D_q = \lim_{w\to 0}\frac{I_q(\Im, w)}{\ln[w]} = \lim_{w\to 0}\frac{\ln\left[\sum_{i=1}^{N_{C_w}^{\Im}} p_i^q(w)\right]}{(q-1)\times\ln[w]}. \tag{6.43}$$

СУФР D_q (6.43) та спектр сингулярностей (мультифрактальний спектр) $f(\alpha)$ є взаємооднозначно пов'язаними прямим (6.36)–(6.37) і зворотним перетвореннями Лежандра [220, 236]:

$$\begin{cases} q = \dfrac{df(\alpha)}{d\alpha}; \\ \tau(q) = \alpha\times\dfrac{df(\alpha)}{d\alpha} - f(\alpha), \end{cases}$$

де

$$\tau(q) = \lim_{w\to 0}\frac{\ln[Z(q,w)]}{\ln[w]} \tag{6.44}$$

— масштабуючий показник Реньї [221], що визначає значення узагальненої статистичної суми $Z(q,w)$ при зменшенні розмірів комірок C_w^i $(w\to 0)$, $i\in\left[1; N_{C_w}^{\Im}\right]$.

МФС $f(\alpha)$ може бути інтерпретований як спектр розмірностей Гаусдорфа-Безіковича D_{HB} компонент заданого зображення $\boldsymbol{I}_{x,y}$, що мають однакові ймовірності заповнення окремих комірок — $p(w)\propto w^{\alpha}$ [220].

На основі аналізу СУФР D_q (6.43) можна визначити наступні статистичні характеристики зображення $\boldsymbol{I}_{x,y}$ [220, 236, 247]: інформаційну розмірність D_1, кореляційну розмірність D_2, граничні узагальнені фрактальні розмірності $D_{-\infty}(D_{+\infty})$.

Використання зазначених характеристик СУФР D_q при проведенні ПСА ЦЗ дає можливість досліджувати обумовлені приховуванням повідомлень до ОПЦЗК зміни ступеня «випадковості» (стохастичності) зображення (інформаційна розмірність D_1) та ступеня кореляції обраних навмання його пікселів (кореляційна розмірність D_2).

Також перспективним є використання граничних узагальнених фрактальних розмірностей $D_{-\infty}(D_{+\infty})$, що відповідають РГБ для компонент графічного контейнера з найменшою (найбільшою) середньою яскравістю значень пікселів в окремих блоках розбиття ЦЗ, для виявлення стеганограм, сформованих з використанням СР матриць яскравості пікселів ЦЗК згідно методу Агарвала (табл. 6.1).

Інформаційна розмірність D_1 відповідає випадку, коли ентропія Реньї (6.42) зводиться до ентропії Шеннона $S(\boldsymbol{I}_{x,y})$ і визначає кількість інформації, необхідної для встановлення значення яскравості пікселя в деякому блоці $C_w(\boldsymbol{x})$:

$$I_q(\boldsymbol{I}_{x,y}, w) = \begin{cases} \dfrac{1}{1-q} \cdot \ln\left[\sum\limits_{i=1}^{N_{C_w}^{\Im}} p_i^q(w)\right], & q \neq 1, \\ -\sum\limits_{i=1}^{N_{C_w}^{\Im}} \{p_i(w) \times log_2[p_i(w)]\}, & q = 1, \end{cases} \tag{6.45}$$

$$S(\boldsymbol{I}_{x,y}) = -\sum_{i=1}^{N_{C_w}^{\Im}} \{p_i(w) \times log_2[p_i(w)]\}, \quad \sum_{i=1}^{N_{C_w}^{\Im}} p_i(w) = 1.$$

Щільність імовірності (ВКФ) $C(r)$ знаходження двох випадковим чином обраних на відстані r значень яскравості пікселів може бути розрахована згідно формули [220]:

$$C(r) = \int_{\Im} \alpha(w, \boldsymbol{x}) \times \alpha(w, \boldsymbol{x} + r)\, dr,$$

де $\alpha(w, \boldsymbol{x}) = \ln[p_i(w)]/\ln(w)$, $i \in \left[1; N_{C_w}^{\Im}\right]$ — відносна щільність розподілу значень яскравості пікселів ЦЗ $\boldsymbol{I}_{x,y}$ у блоці C_w^i; $\boldsymbol{x}$ ($\boldsymbol{x} \in \mathbb{R}^2$) — центр окремого блока (точка перетину діагоналей).

Тоді кореляційна розмірність D_2 — оцінка ймовірності того, що відстань r між двома випадковим чином обраними значеннями яскравості пікселів зображення $\boldsymbol{I}_{x,y}$ є меншою за w — розраховується згідно виразу [220]:

$$C(r) \propto r^{-(D_T - D_2)},$$

де D_T — розмірність Лебега (*Lebesgue covering dimension*) простору, в який вкладено область $\Im$ (у випадку аналізу зображень — $D_T = 2$).

Для розрахунку СУФР D_q і МФС $f(\alpha)$ згідно формул (6.43) та (6.36), (6.37) необхідним є визначення розподілу значень яскравості пікселів зображення при варіації розміру блоків розбиття C_w^i, $i \in \left[1; N_{C_w}^{\Im}\right]$ у широких межах, що є обчислювально складною процедурою. Тому при практичному застосуванні МФА широко використовуються швидкі методи визначення D_q та $f(\alpha)$ — наприклад, метод *BCM* (*Box-Counting Method*) [251], метод ізаритм[XXXIX] [252], метод покриттів [160] тощо.

[XXXIX] Ізаритма (*isarithmic line*; *isarithm*) — лінія, що з'єднує точки з однаковими статистичними показниками.

В останній час все більшої популярності набувають методи оцінки РГБ та узагальнених фрактальних розмірностей D_q, засновані на використанні вейвлет перетворення досліджуваного сигналу [230, 253], наприклад, вже згаданий вище метод *WTMM* [254], метод найбільших коефіцієнтів вейвлет-перетворення (*Wavelet Leaders Method*) [255]. Незважаючи на універсальність такого підходу, його практичне застосування для обробки складних сигналів потребує проведення декількох вейвлет-перетворень досліджуваного сигналу з різними базисними функціями для забезпечення високої точності визначення узагальнених фрактальних розмірностей D_q, що є обчислювально затратною процедурою.

Одними з найбільш поширених підходів до визначення РГБ цифрових зображень є клас методів *DBCM* (*Differential Box-Counting Method*) [256], особливістю яких є представлення досліджуваного півтонового зображення як тривимірної поверхні — залежності значення яскравості пікселів від їх просторового положення. Згідно *DBCM* для визначення розподілу значень висот у блоках розбиття C_w^i, $i \in \left[1; N_{C_w}^{\Im}\right]$ використовується ковзне вікно розміром $w \times w$ пікселів, що дозволяє забезпечити високу точність оцінки РГБ при низькій обчислювальній складності методів обробки.

Відомим обмеженням *DBCM* є залежність розрахованого значення РГБ D_{HB} від вибору розмірів w блоків розбиття [251]. Для подолання даного обмеження був запропонований вдосконалений метод *DBCM* (*Improved DBCM, IDBCM*) [257], заснований на розбитті півтонового зображення $\boldsymbol{I}_{x,y}$ на блоки розміром $w \times w$ та величиною перекриття суміжних блоків Δ_w рядків (стовпчиків). Для визначення СУФР D_q і МФС $f(\alpha)$ нами була проведена модифікація *IDBCM*, що полягала у введенні додаткового етапу обробки ЦЗ — розрахунку узагальненої статистичної суми $Z(w, q)$ (6.41) та ентропії Реньї I_q (6.45) порядку q для розподілу висот блоків розбиття C_w^i, $i \in \left[1; N_{C_w}^{\Im}\right]$.

Обробка кольорового зображення $\boldsymbol{I}_{x,y}$ розміром $M \times N$ пікселів з використанням модифікованого методу *IDBCM* проводиться в декілька етапів [257]. На першому етапі матриці яскравості пікселів кожного каналу кольору заданого зображення розбиваються на блоки B_w розміром $w \times w$ пікселів, що перекриваються (величина перекриття — Δ_w рядків (стовпчиків)). Для кожного блока B_w яскравості пікселів зображення $\boldsymbol{I}_{x,y}$ розраховується відносна висота $n_w(i,j)$ [257]:

$$n_w(i,j) = B_w^{max}(i,j) - B_w^{min}(i,j) + 1;$$
$$i \in [1; \lfloor M/w \rfloor], \; j \in [1; \lfloor N/w \rfloor], \tag{6.46}$$

де

$$B_w^{max}(i,j) = \left\lceil \frac{\max\{B_w(i,j)\}}{\acute{w}_{opt}} \right\rceil, \quad B_w^{min}(i,j) = \left\lceil \frac{\min\{B_w(i,j)\}}{\acute{w}_{opt}} \right\rceil$$

— відповідно, номера блоків, до яких потрапляють максимальне й мінімальне значення яскравості пікселів для поточного блока розбиття B_w;

$$\acute{w}_{opt} = \left\lceil \frac{w}{1+2\sigma} \right\rceil, \quad \sigma = \sqrt{D[\boldsymbol{I}_{x,y}]}$$

— поточне значення кроку розбиття блоків B_w по висоті; σ — середньоквадратичне відхилення значень яскравості пікселів ЦЗ $\boldsymbol{I}_{x,y}$.

Розбиття ЦЗ повторюється декілька разів (але не менше трьох) при варіації величини блока розбиття w. За результатами дослідження [257] встановлено, що мінімальна похибка визначення значення РГБ досягається при величині перекриття блоків B_w рівній $\Delta_w = 1$.

На другому етапі проводиться розрахунок узагальненої статистичної суми $Z(w,q)$ й ентропії Реньї I_q порядку q згідно виразів (6.41) і (6.45) при заміні ймовірності $p(w)$ на значення нормованого розподілу відносних висот N_w^{norm} блоків розбиття B_w:

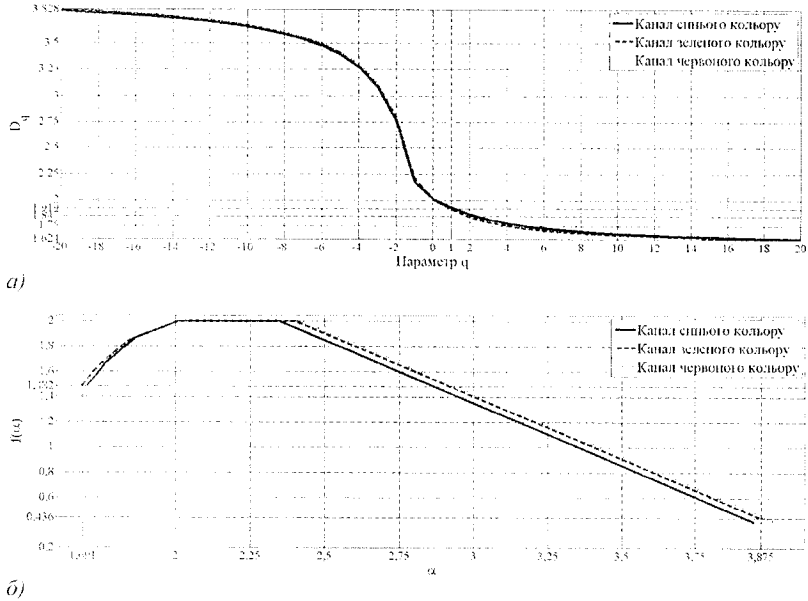
$$N_w^{norm} = \frac{n_w(i,j)}{\sum_{i,j} n_w(i,j)}.$$

На третьому етапі проводиться побудова залежності ентропії Реньї I_q від розмірів блоків розбиття $(1/w)$ у логарифмічних координатах — $lg[I_q] = f(lg[1/w])$. Значення узагальненої фрактальної розмірності D_q визначається як тангенс кута нахилу прямої, отриманої шляхом лінійної апроксимації одержаної залежності. Для побудови СУФР D_q другий і третій етапи обробки каналів кольору зображення $\boldsymbol{I}_{x,y}$ згідно *IDBCM* методу повторюються при варіації масштабуючого параметра q ($q \in \mathbb{R}$).

На четвертому етапі розраховується МФС $f(\alpha)$ шляхом застосування до отриманого СУФР D_q прямого перетворення Лежандра (6.36), (6.37).

Для визначення граничних значень параметра q, аналогічно до флуктуаційного аналізу (див. підп. 6.6.2.2), проводиться мультифрактальний аналіз зображень з тестового пакета (табл. 6.2) при варіації мінімального q_{min} та максимального q_{max} значення параметра q з кроком $\Delta_q = 0{,}5$. За результатами аналізу отриманих даних встановлено, що зміни граничних фрактальних розмірностей $D_{-\infty}(D_{+\infty})$ при збільшенні абсолютних значень q ($|q| > 20$) не перевищують 0,01. Тому при проведенні МФА каналів кольору ЦЗ значення масштабуючого параметра q змінювалися в діапазоні $q \in [-20; 20]$ з кроком $\Delta_q = 0{,}5$.

На рис. 6.13 наведено СУФР D_q та МФС $f(\alpha)$, розраховані згідно модифікованого методу *IDBCM* для псевдовипадково обраного тестового зображення (табл. 6.2).



а)

б)

Рис. 6.13. Спектри узагальнених фрактальних розмірностей D_q *(а)* та відповідні мультифрактальні спектри $f(\alpha)$ *(б)*, обчислені для окремих каналів кольору тестового зображення $\boldsymbol{I}_{x,y}$ згідно модифікованого методу *IDBCM*

Спектри узагальнених фрактальних розмірностей D_q для окремих каналів кольору ЦЗ (рис. 6.13, *а*) практично не відрізняються один від одного, що свідчить про близькість розподілу значень яскравості пікселів у кожному каналі кольору.

На основі виразу (6.44) можна встановити прямий зв'язок між граничними фрактальними розмірностями $D_{-\infty}(D_{+\infty})$ і значеннями локальних масштабуючих показників α [236]:

$$\left.\frac{d\tau(q)}{dq}\right|_{q\to(+\infty)} = D_{+\infty} = \alpha_{min}, \qquad \left.\frac{d\tau(q)}{dq}\right|_{q\to(-\infty)} = D_{-\infty} = \alpha_{max}.$$

За результатами проведених розрахунків були визначені як граничні фрактальні розмірності $D_{-\infty}$ $(D_{+\infty})$, так і відносні щільності α_{max} (α_{min})

розподілу значень яскравості пікселів окремих каналів кольору зображення (рис. 6.13):

$$D_{-\infty}^{red} = 3.828,\ \alpha_{max}^{red} = 3.912,\quad D_{+\infty}^{red} = 1.624,\ \alpha_{min}^{red} = 1.691;$$

$$D_{-\infty}^{green} = 3.822,\ \alpha_{max}^{green} = 3.975,\quad D_{+\infty}^{green} = 1.623,\ \alpha_{min}^{green} = 1.690;$$

$$D_{-\infty}^{blue} = 3.823,\ \alpha_{max}^{blue} = 3.860,\quad D_{+\infty}^{blue} = 1.627,\ \alpha_{min}^{blue} = 1.705.$$

Відмінність у значеннях $D_{-\infty}$ $(D_{+\infty})$ та α_{max} (α_{min}) пояснюється використанням обмеженого інтервалу значень параметра q $(q \in [-20; 20])$.

Для визначення фрактальних характеристик цифрових зображень, на основі модифікованого методу *IDBCM* у [161] представлено алгоритм обробки каналів кольору ЦЗ, що дозволяє розраховувати СУФР D_q та МФС (α), а також їхні основні характеристики:

1) значення інформаційної D_1 та кореляційної D_2 розмірностей;

2) граничні узагальнені фрактальні розмірності $D_{-\infty}(D_{+\infty})$;

3) середні $(E[\acute{D}_q])$ та максимальні $(\max[\acute{D}_q])$ значення похідних СУФР при зміні параметра q в діапазонах $q \in [q_{min}; 0]$ та $q \in [0; q_{max}]$;

4) максимальне $(f^{max}(\alpha),\ \alpha_{max})$ й мінімальне $(f^{min}(\alpha), \alpha_{min})$ значення розмірностей Гаусдорфа-Безіковича та ймовірностей заповнення блоків розбиття ЦЗ для його монофрактальних компонент;

5) середні $(E[\acute{f}(\alpha)])$ та максимальні $(\max[\acute{f}(\alpha)])$ значення похідних мультифрактального спектра $f(\alpha)$.

Для визначення ймовірності виявлення стеганограм з даними, вбудованими до ОПЦЗК згідно одноетапних, багатоетапних і комплексних методів, на основі алгоритму [161] був розроблений стеганодетектор SD_{MFA}. Загальна кількість фрактальних характеристик зображень-контейнерів, використаних при формуванні кластера ознак стеганограм, склала $d_{MFA} = 14$. Налаштування SD_{MFA} було проведено аналогічно до налаштування стеганодетектора SD_{VgA} із застосуванням ансамблю класифікаторів (див. підп. 6.6.2.1) — лінійних дискримінантів Фішера. Визначення параметрів АК (кількості базових класифікаторів і потужності множини ознак кожного з них), налаштування АК та ЛДФ проводилося згідно алгоритмів, наведених у [161].

Виявлення стеганограм, сформованих згідно одноетапних методів Дея та Агарвала, багатоетапних методів Джозефа та Хана, а також комплексних методів Елахіана та Ганджела, проводилося з використанням SD_{MFA} на пакеті з 2500 тестових цифрових зображень (табл. 6.2). Встановлено, що сімейства залежностей значень метрики

AUC від ступеня заповнення ЦЗК стеганоданими Δ_C та вагового коефіцієнта G для стеганодетектора SD_{MFA} при вбудовуванні СД типу «Креслення», «Карта» й «Портрет» (табл. 6.2) до різних каналів кольору ЦЗК згідно одноетапних, багатоетапних і комплексних методів, мають однаковий характер та несуттєво різняться діапазоном значень метрики AUC [161].

В якості прикладу, на рис. 6.14 наведені сімейства залежностей значень метрики AUC від ступеня заповнення ЦЗК стеганоданими Δ_C та вагового коефіцієнта G для стеганодетектора SD_{MFA} при вбудовуванні СД типу «Портрет» до каналу синього кольору ЦЗК згідно одноетапних, багатоетапних і комплексних методів.

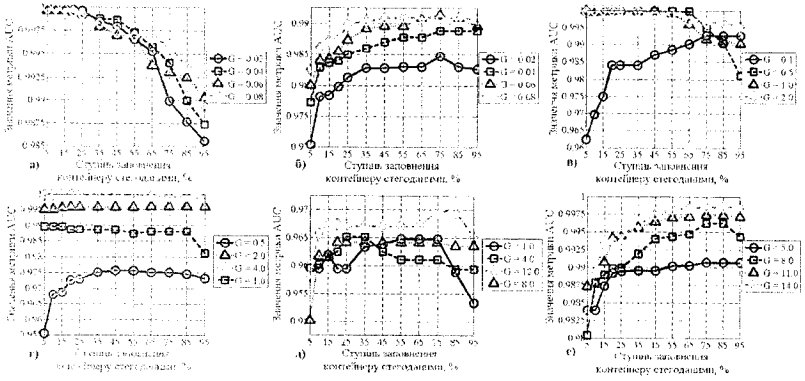


Рис. 6.14. Сімейства залежностей значень метрики AUC від ступеня заповнення зображення-контейнера стеганоданими і вагового параметра G, одержані для стеганодетектора SD_{MFA} при вбудовуванні СД типу «Портрет» до каналу синього кольору ЦЗК згідно: *(а)* – методу Дея; *(б)* – методу Аґарвала; *(в)* – методу Джозефа; *(г)* – методу Хана; *(ґ)* – методу Елахіана; *(е)* – методу Ґанджела

Використання МФА для виявлення стеганограм, сформованих згідно одноетапних методів Дея та Аґарвала, а також двоетапного методу Джозефа приховання повідомлень до ОПЦЗК, дало можливість збільшити ймовірність виявлення факту приховання повідомлень (метрика AUC) в області слабкого заповнення ЦЗК стеганоданими ($\Delta_C = 5\%$) та мінімальній енергії СД ($G = G_{min}$) у порівнянні з результатами, отриманими для варіограмного (рис. 6.9) й мультифрактального флуктуаційного (рис. 6.12) аналізів.

У випадку використання багатоетапного методу Хана та комплексних методів Елахіана та Ґанджела (рис. 6.14, *г–е*) формування стегано-

грам значення метрики *AUC* для мультифрактального аналізу останніх є співставними з відповідними результатами для статистичних моделей *SPAM* (рис. 6.3), *CC-PEV* (рис. 6.4) та *CDF* (рис. 6.5), використовуючи, проте, простір ознак суттєво меншої розмірності — табл. 6.9.

У табл. 6.10 наведено значення метрики *AUC* при використанні методів структурного й статистичного аналізу сформованих згідно методу Агарвала стеганограм, у найбільш складному випадку ПСА — слабкого заповнення ЦЗК стеганоданими ($\Delta_C = 5\%$) та малій енергії прихованих повідомлень ($G = G_{min}$).

Таблиця 6.9

Кількість елементів простору ознак стеганодетектора, налаштованого з використанням методів структурного й статистичного аналізу зображень, а також універсального стеганодетектора Авсібаша

Тип використовуваного простору ознак	Кількість елементів
Кореляційні характеристик зображень (варіограмний аналіз)	39
Кореляційні й фрактальні характеристики зображень (мультифрактальний флуктуаційний аналіз)	182
Фрактальні характеристики зображень (мультифрактальний аналіз)	14
Статистична модель *SPAM*	686
Статистична модель *CC-PEV*	548
Статистична модель *CDF*	1234
Метрики якості цифрових зображень (УСД Авсібаша)	14

Таблиця 6.10

Значення *AUC* при використанні методів структурного й статистичного аналізу сформованих згідно методу Агарвала стеганограм, у випадку слабкого рівня заповнення ЦЗК ($\Delta_C = 5\%$) та мінімальній енергії прихованих повідомлень

	Статистична модель *CDF*	Методи структурного аналізу ЦЗ:		
		ВА	МФФА	МФА
Метод Дея	0.875	0.729	0.977	0.986
Метод Агарвала	0.774	0.665	0.910	0.971
Метод Джозефа	0.603	0.770	0.969	0.962
Метод Хана	0.999	0.712	0.937	0.955
Метод Елахіана	0.999	0.600	0.757	0.950
Метод Ганджела	0.999	0.668	0.975	0.980

Застосування МФА цифрових зображень дозволяє забезпечити високу ймовірність виявлення стеганограм (метрика AUC) у найбільш складних випадках ПСА — слабкого заповнення ЦЗК стеганоданими ($\Delta_C = 5\%$) й використанні спеціальних перетворень ЦЗК, зокрема СР, — коли використання відомих методів ПСА (наприклад, статистичного стеганоаналізу) є неефективним. Також перевагою використання МФА при проведені ПСА ЦЗ є відносно малий час визначення характеристик стеганограм з даними, вбудованими до ОПЦЗК — $\langle T_{MFA} \rangle \approx 7.5$ с, — що є співставним з часом обробки зображень з використанням ВА, коли $\langle T_{VgA} \rangle \approx 7.2$ с (див. табл. 6.7).

У порівнянні з методами статистичного стеганоаналізу використання кореляційних і фрактальних характеристик ЦЗ при проведенні ПСА дозволяє суттєво підвищити точність виявлення стеганограм, сформованих згідно одноетапних, багатоетапних і комплексних методів. Утім, величина «впливу» зазначених характеристик цифрових зображень на ймовірність виявлення стеганограм (метрика AUC) є неоднаковою. В якості прикладу на рис. 6.15 наведені ROC-криві залежності частки виявлених стеганограм (*True Positive Rate*, TPR), сформованих згідно методу Дея, від значень помилок першого роду (*False Positive Rate*, FPR) при використанні груп кореляційних (параметри СУЕГ h_q) і фрактальних (параметри МФС $f_q(\alpha_q)$) характеристик шумових компонент ЦЗ, розрахованих із застосуванням МФФА.

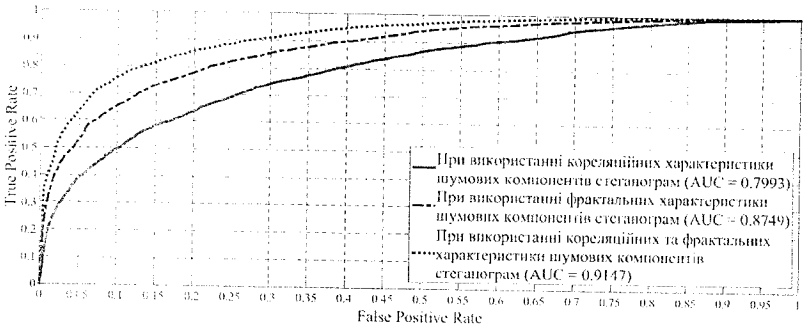


Рис. 6.15. Залежність кількості виявлених стеганограм від значень помилок першого роду стеганодетектора при використанні груп кореляційних і фрактальних характеристик шумових компонент стеганограм, розрахованих із застосуванням МФФА

Побудова ROC-характеристики стеганодетектора проводиться згідно наступних формул [184, 258]:

$$FPR(T) = \int_{T}^{+\infty} P_0(T)dT; \qquad TPR(T) = \int_{T}^{+\infty} P_1(T)dT,$$

де P_0, P_1 — відповідно, ймовірності віднесення стеганограм до класу пустих і заповнених зображень-контейнерів; T — порогове значення для вирішуючого правила стеганодетектора (наприклад, відношення правдоподібності для ЛДФ).

Об'єднання кореляційних і фрактальних характеристик шумових компонент ЦЗ при проведенні ПСА призводить до нелінійного збільшення величини TPR (рис. 6.15), що свідчить про нерівномірний «вплив» зазначених характеристик на значення ймовірності виявлення стеганограм (метрика AUC). Визначення кореляційних і фрактальних характеристик зображення-контейнера, що мають найбільший «вплив» на точність виявлення стеганограм, дозволить зменшити розмірність простору ознак структурних стеганодетекторів та, відповідно, час їхнього налаштування при забезпеченні фіксованої ймовірності виявлення стеганограм.

6.7. ФОРМУВАННЯ КЛАСТЕРА ДЕМАСКУЮЧИХ ОЗНАК СТЕГАНОГРАМ

Використання методів структурного аналізу цифрових зображень — варіограмного, флуктуаційного й мультифрактального — дозволяє підвищити точність виявлення стеганограм з даними, вбудованими до ОПЦЗК із застосуванням спектральних (двовимірного дискретного вейвлет перетворення) і спеціальних (сингулярний розклад) перетворень ЦЗК, у порівнянні зі статистичними та універсальними стеганодетекторами. Проте величина впливу кореляційних (N, S та R-параметрів ЦЗ, параметрів спектра узагальнених експонент Герста) і фрактальних (параметрів спектра узагальнених фрактальних розмірностей та мультифрактального спектра) характеристик стеганограм на значення ймовірності виявлення прихованих повідомлень є неоднаковою. Тому важливою задачею є формування кластера демаскуючих ознак (ДМО) стеганограм — визначення саме тих кореляційних і фрактальних характеристик ЦЗ, що найбільше змінюються внаслідок приховання повідомлень до ОПЦЗК — для одноетапних методів Дея та Аґарвала, а також двоетапного методу Джозефа. Це дозволить скоротити розмірність простору ознак структурних стеганодетекторів SD_{VgA}, SD_{MFDFA} і SD_{MFA} та, відповідно, спростити процедуру налаштування стеганодетектора при забезпеченні фіксованої точності виявлення стеганограм.

6.7.1. Варіограмний аналіз стеганограм

Для визначення кластера ДМО стеганограм, сформованих згідно одноетапних методів Дея та Агарвала, багатоетапних методів Джозефа та Хана, а також комплексних методів Елахіана та Ганджела, на основі пакета з 2500 тестових зображень (табл. 6.2) було проведено дослідження кореляційних характеристик ЦЗК (N, S та R-параметрів) та стеганограм при варіації рівня заповнення ЦЗК стеганоданими Δ_C та вагового параметра G.

Для порівняння величини змін кореляційних характеристик ЦЗК, обумовлених приховуванням повідомлень до ОПЦЗК, було використано величину відносного відхилення Δ_ε:

$$\Delta_\varepsilon = \left| \frac{a_{cover} - a_{stego}}{a_{cover}} \right| \times 100\%,$$

де a_{cover}, a_{stego} — відповідно, значення параметра a для зображення-контейнера та стеганограми.

Значення N, S та R-параметрів для зображень-контейнерів і стеганограм, сформованих згідно одноетапних методів Дея і Агарвала, а також двоетапного методу Джозефа, при слабкому заповненні ЦЗК стеганоданими ($\Delta_C = 10\%$) та мінімальній енергії приховуваних повідомлень ($G = G_{min}$) наведені у табл. 6.11–6.13.

Таблиця 6.11

Характеристики розподілу значень N-параметра для каналу синього кольору ЦЗК і стеганограм, сформованих згідно одноетапних методів Дея і Агарвала, а також двоетапного методу Джозефа, при слабкому заповненні ЦЗК СД ($\Delta_C = 10\%$) і мінімальній енергії прихованих повідомлень ($G = G_{min}$)

Характеристики розподілу значень	ЦЗК	Стеганограма:					
		за методом Дея		за методом Агарвала		за методом Джозефа	
		значення	Δ_ε, %	значення	Δ_ε, %	значення	Δ_ε, %
Середнє значення	185.09	177.71	3.99	186.15	0.57	191.33	3.37
Дисперсія	16219.61	15000.82	**7.51**	17305.51	**6.70**	17351.90	**6.98**
Коефіцієнт асиметрії	0.82	0.82	0.00	0.83	1.16	0.82	0.11
Коефіцієнт ексцесу	0.22	0.22	0.44	0.26	**17.31**	0.22	0.65
1-й квартиль, $F_{0.25}$	68.37	65.77	3.81	69.96	2.33	70.54	3.17
Медіана (квартиль $F_{0.50}$)	144.73	139.12	3.88	143.75	0.68	148.17	2.37
3-й квартиль, $F_{0.75}$	254.60	244.38	4.01	261.85	2.85	262.61	3.15
Інтерквартильний розмах	186.23	150.85	**19.00**	165.29	**11.24**	162.22	**12.89**
Мода розподілу	8.52	8.04	**5.56**	7.03	**17.50**	6.97	**18.16**
Розмах варіації	583.36	560.52	3.92	594.45	1.90	604.76	3.67
Коефіцієнт осциляцій	2.94	2.94	0.01	2.95	0.36	2.95	0.22
Коефіцієнт варіації	0.70	0.70	0.01	0.70	0.12	0.70	0.34
Лінійний коефіцієнт варіації	65.53	63.36	3.32	69.69	**6.35**	67.40	2.84

Таблиця 6.12

Характеристики розподілу значень S-параметра для каналу синього кольору ЦЗК і стеганограм, сформованих згідно одноетапних методів Дея і Агарвала, а також двоетапного методу Джозефа, при слабкому заповненні ЦЗК СД ($\Delta_C = 10\%$) і мінімальній енергії прихованих повідомлень ($G = G_{min}$)

Характеристики розподілу значень	ЦЗК	Стеганограма:					
		за методом Дея		за методом Агарвала		за методом Джозефа	
		значення	Δ_ε, %	значення	Δ_ε, %	значення	Δ_ε, %
Середнє значення	6230.61	6012.40	3.50	6367.93	2.20	6269.78	0.63
Дисперсія	15477498	14300167	**7.61**	16424073	**6.12**	15640561	1.05
Коефіцієнт асиметрії	0.70	0.70	0.04	0.71	1.69	0.69	0.62
Коефіцієнт ексцесу	-0.11	-0.12	4.65	-0.11	3.21	-0.12	**5.38**
1-й квартиль, $F_{0.25}$	2720.61	2609.03	4.10	2761.80	1.51	2746.14	0.94
Медіана (квартиль $F_{0.50}$)	5279.62	5058.81	4.18	5403.84	2.35	5303.78	0.46
3-й квартиль, $F_{0.75}$	8555.30	8242.09	3.66	8763.04	2.43	8585.26	0.35
Інтерквартильний розмах	5834.69	4621.72	**20.79**	4862.09	**16.67**	4809.21	**17.58**
Мода розподілу	325.26	312.74	3.85	329.45	1.29	327.14	0.58
Розмах варіації	16192.90	15582.74	3.77	16771.17	3.57	16340.83	0.91
Коефіцієнт осциляції	2.56	2.56	0.05	2.58	0.49	2.56	0.00
Коефіцієнт варіації	0.66	0.66	0.36	0.66	0.28	0.66	0.31
Лінійний коефіцієнт варіації	2051.18	1964.82	4.21	2084.98	1.65	2054.05	0.14

Таблиця 6.13

Характеристики розподілу значень R-параметра для каналу синього кольору ЦЗК і стеганограм, сформованих згідно одноетапних методів Дея і Агарвала, а також двоетапного методу Джозефа, при слабкому заповненні ЦЗК СД ($\Delta_C = 10\%$) і мінімальній енергії прихованих повідомлень ($G = G_{min}$)

Характеристики розподілу значень	ЦЗК	Стеганограма:					
		за методом Дея		за методом Агарвала		за методом Джозефа	
		значення	Δ_ε, %	значення	Δ_ε, %	значення	Δ_ε, %
Середнє значення	148.05	149.26	0.82	149.62	1.06	148.69	0.43
Дисперсія	3993.69	3990.46	0.08	3934.28	1.49	3999.51	0.15
Коефіцієнт асиметрії	-0.06	-0.08	**28.06**	-0.08	**30.23**	-0.06	2.12
Коефіцієнт ексцесу	-1.03	-1.04	0.65	-1.04	0.64	-1.04	0.76
1-й квартиль, $F_{0.25}$	78.00	78.00	0.00	78.00	0.00	78.00	0.00
Медіана (квартиль $F_{0.50}$)	142.00	142.00	0.00	142.00	0.00	142.00	0.00
3-й квартиль, $F_{0.75}$	209.00	212.00	1.44	212.00	1.44	209.00	0.00
Інтерквартильний розмах	131.00	116.00	**11.45**	116.00	**11.45**	116.00	**11.45**
Мода розподілу	78.00	78.00	0.00	78.00	0.00	78.00	0.00
Розмах варіації	176.00	176.00	0.00	176.00	0.00	176.00	0.00
Коефіцієнт осциляції	1.38	1.35	1.66	1.35	1.66	1.37	0.54
Коефіцієнт варіації	0.44	0.44	0.64	0.44	1.27	0.44	0.12
Лінійний коефіцієнт варіації	30.50	30.25	0.82	30.50	0.00	30.00	1.64

Як видно з табл. 6.11–6.13, найбільших змін при вбудовуванні СД до ОПЦЗК з використанням як спектральних (двовимірного дискретного вейвлет перетворення), так і спеціальних (сингулярний розклад) перетворень ЦЗК згідно методів Дея, Агарвала та Джозефа, зазнають наступні характеристики розподілу значень *N*, *S* та *R*-параметрів по рядках зображень-контейнерів:

1) *N*-параметр цифрових зображень:

a) дисперсія (від $\Delta_\varepsilon = 6{,}70\%$ — для методу Агарвала до $\Delta_\varepsilon = 7{,}51\%$ — для методу Дея);

b) інтерквартильний розмах (від $\Delta_\varepsilon = 11{,}24\%$ — для методу Агарвала до $\Delta_\varepsilon = 19{,}00\%$ — для методу Дея);

c) мода розподілу (від $\Delta_\varepsilon = 5{,}56\%$ — для методу Дея до $\Delta_\varepsilon = 18{,}16\%$ — для методу Джозефа);

2) *S*-параметр цифрових зображень: інтерквартильний розмах (від $\Delta_\varepsilon = 16{,}67\%$ — для методу Агарвала до $\Delta_\varepsilon = 20{,}79\%$ — для методу Дея);

3) *R*-параметр цифрових зображень: інтерквартильний розмах ($\Delta_\varepsilon \approx 11{,}45\%$ для методів Дея, Агарвала та Джозефа).

Зміни інших характеристик значень *N*, *S* та *R*-параметрів ЦЗК — середнього та медіанного значень, коефіцієнтів асиметрії та ексцесу, першого й третього квартилів, розмаху варіації, коефіцієнтів варіації та осциляції, а також лінійного коефіцієнту варіації не перевищують 5% (див. табл. 6.11–6.13).

Для оцінки ймовірності виявлення стеганограм з даними, вбудованими до ОПЦЗК згідно одноетапних, багатоетапних і комплексних методів, при використанні скороченого переліку кореляційних характеристик ЦЗ було проведено модифікацію стеганодетектора SD_{VgA}. Налаштування модифікованого стеганодетектора SD_{VgA}^{mod} проводилося аналогічно до SD_{VgA} з використанням ансамблю класифікаторів згідно алгоритму [161].

Порівняння точності виявлення стеганограм з даними, вбудованими до ОПЦЗК, при використанні стеганодетекторів SD_{VgA} і SD_{VgA}^{mod} проводилося на тестовому пакеті з 2500 цифрових зображень (табл. 6.2). Значення метрики AUC і середня тривалість обробки зображень $\langle T \rangle$ при застосуванні SD_{VgA} і SD_{VgA}^{mod}, статистичного стеганодетектора SD_{CDF}, а також УСД Авсібаша для виявлення СД, вбудованих до канал синього кольору ЦЗК згідно одноетапних, багатоетапних і комплексних методів, у випадках слабкого (AUC_{min}, $\Delta_C = 5\%$) при $G = G_{min}$ та сильного (AUC_{max}, $\Delta_C = 85\%$) при $G = G_{max}$ заповнення ЦЗК стеганоданими наведені у табл. 6.14.

Таблиця 6.14

Значення AUC та середній час обробки $\langle T \rangle$ ЦЗ при застосуванні стеганодетекторів SD_{VgA} і SD_{VgA}^{modif}, статистичного стеганодетектора SD_{CDF} та УСД Авсібаша (гаусівська фільтрація стеганограм) для виявлення СД, вбудованих до каналу синього кольору ЦЗК згідно одноетапних, багатоетапних і комплексних методів, у випадку слабкого (AUC_{min}) і сильного (AUC_{max}) заповнення ЦЗК

		Статистична модель CDF	УСД Авсібаша	Варіограмний аналіз:	
				SD_{VgA}	SD_{VgA}^{modif}
Метод Дея	AUC_{max} / AUC_{min}	0.999 / 0.725	0.915 / 0.551	0.999 / 0.778	0.990 / 0.731
	$\langle T \rangle$, сек	11.4	5.3	7.1	6.5
Метод Агарвала	AUC_{max} / AUC_{min}	0.993 / 0.724	0.646 / 0.514	0.999 / 0.665	0.961 / 0.640
	$\langle T \rangle$, сек	11.8	5.8	7.4	6.7
Метод Джозефа	AUC_{max} / AUC_{min}	0.999 / 0.603	0.859 / 0.507	0.999 / 0.769	0.989 / 0.701
	$\langle T \rangle$, сек	11.3	5.5	7.2	6.6
Метод Хана	AUC_{max} / AUC_{min}	0.999 / 0.997	0.999 / 0.997	0.999 / 0.703	0.999 / 0.698
	$\langle T \rangle$, сек	11.2	5.2	7.1	6.6
Метод Елахіана	AUC_{max} / AUC_{min}	0.999 / 0.991	0.720 / 0.713	0.999 / 0.774	0.999 / 0.763
	$\langle T \rangle$, сек	11.0	5.3	7.2	6.5
Метод Ганджела	AUC_{max} / AUC_{min}	0.999 / 0.998	0.943 / 0.671	0.999 / 0.668	0.999 / 0.641
	$\langle T \rangle$, сек	11.1	5.2	7.1	6.6

Скорочення переліку кореляційних характеристик ЦЗ (N, S та R-параметрів) на 87% — від 39 (SD_{VgA}) до 5 (SD_{VgA}^{modif}) параметрів — не призвело до суттєвого зниження значень імовірності виявлення стеганограм (метрика AUC), сформованих згідно одноетапних, багатоетапних та комплексних методів, у випадку як сильного ($\Delta AUC_{max} \leq 0.038$) так і слабкого ($\Delta AUC_{min} \leq 0.068$) заповнення ЦЗК стеганоданими (див. табл. 6.14).

Перевагою використання скороченого переліку кореляційних характеристик ЦЗ є зменшення середнього часу обробки зображень на 8% — від 7.1 с до 6.6 с. Внаслідок цього можна рекомендувати використовувати модифікований стеганодетектор SD_{VgA}^{modif} для попереднього віднесення ЦЗ до класів пустих чи заповнених зображень-контейнерів (стеганограм) без необхідності використання складних статистичних стеганодетекторів.

Формування стеганограм згідно методів Агарвала та Джозефа з використанням декількох компонент ЦЗК, що відповідають різним сингулярним числам λ_i, $i \in [1; N]$ зображення-контейнера (6.6), призводить до суттєвого зниження точності виявлення стеганограм при використанні ВА (табл. 6.14). Виявлення сформованих стеганограм потребує використання обчислювано складного мультифрактального флуктуаційного аналізу зображень (табл. 6.8). Тому важливою є задача модифікації МФФА — визначення кореляційних і фрактальних характеристик шумових компонентів ЦЗК, що найбільше змінюються при вбудовуванні СД з використанням СР ЦЗК — розв'язок якої дозволить скоротити розмірність простору ознак стеганодетектора SD_{MFDFA} і, відповідно, зменшити час обробки ЦЗ.

6.7.2. Аналіз фрактальних характеристик стеганограм

6.7.2.1. Мультифрактальний флуктуаційний аналіз шумових компонент стеганограм

Для виявлення ДМО стеганограм, сформованих згідно одноетапних, багатоетапних і комплексних методів, було проведено дослідження змін кореляційних і фрактальних характеристик шумових компонент ЦЗК при вбудовуванні СД до ОПЦЗК. За результатами аналізу отриманих даних [237, 259], було встановлено, що найбільших змін при формуванні стеганограм зазнають похідні $h'_q = dh_q/dq$ СУЕГ h_q, а також ширина Δ_{α_q} і розмах Δ_{f_q} МФС $f_q(\alpha_q)$ по рядках ЦЗК.

За результатами дослідження змін значень похідних h'_q при вбудовуванні СД до ОПЦЗК було встановлено, що найбільших змін при формуванні стеганограм зазнає інтерквартильний розмах значень h'_q, а зміни середніх ($\Delta_{mean}(h'_q) \leq 0.05$) та медіанних ($\Delta_{median}(h'_q) \leq 0.06$) значень похідних є відносно малими, незалежно від ступеня заповнення зображення-контейнера стеганоданими Δ_C і значення вагового параметра G (табл. 6.1).

На рис. 6.16, 6.17 наведено залежності інтерквартильного розмаху значень похідних h'_q спектра узагальнених експонент Герста стеганограм, сформованих згідно одноетапних, багатоетапних і комплексних методів, від ступеня заповнення ЦЗК стеганоданими Δ_C при мінімальному ($G = G_{min}$) та максимальному ($G = G_{max}$) значенні вагового параметра G для слабких ($q \in [q_{min}; 0]$) і сильних ($q \in [0; q_{max}]$) флуктуацій значень яскравості пікселів стеганограм.

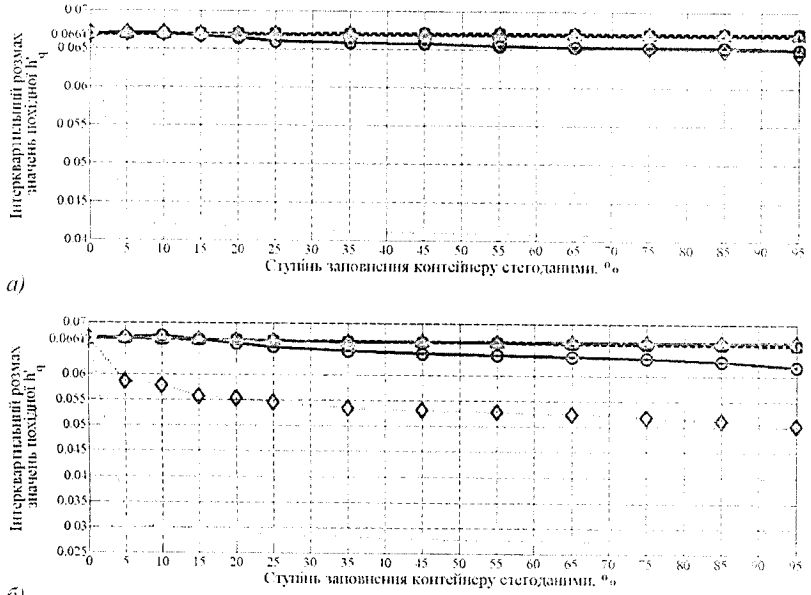
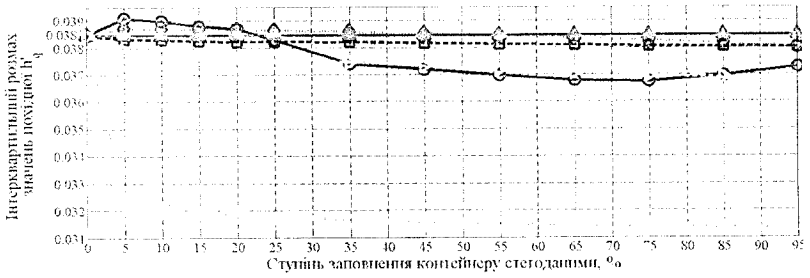


Рис. 6.16. Залежності інтерквартильного розмаху значень похідних h'_q від ступеня заповнення ЦЗК СД Δ_C при мінімальних $G = G_{min}$ *(а)* і максимальних $G = G_{max}$ *(б)* значеннях вагового параметра G для слабких ($q \in [q_{min}; 0]$) флуктуацій значень яскравості пікселів стеганограм

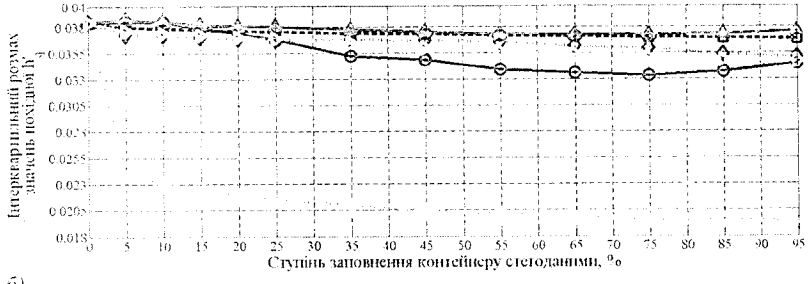
Найбільші зміни інтерквартильного розмаху значень похідних h'_q були виявлені для стеганограм, сформованих згідно багатоетапного методу Хана та комплексного методу Ганджела, у випадку дослідження як слабких (рис. 6.16), так і сильних (рис. 6.17) флуктуацій яскравості пікселів ЦЗ. Величина змін похідної h'_q суттєво залежить від значення вагового параметра G, пропорційного енергії стеганографічних даних, що дає можливість використовувати для визначення наявності прихованих повідомлень, а також оцінки їхньої енергії прості порогові методи.

Демаскуючою ознакою стеганограм, сформованих згідно двоетапного методу Джозефа, є суттєве зниження значення h'_q для слабких флуктуацій значень яскравості пікселів ЦЗК ($q < 0$, рис. 6.16, *б*). Це дозволяє використовувати прості порогові методи для виявлення стеганограм навіть при слабкому заповненні ЗК стеганоданими ($\Delta_C \leq 10\%$).

Формування стеганограм з використанням спеціальних методів обробки ЦЗК згідно методу Агарвала не призводить до суттєвих змін



а)



б)

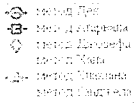


Рис. 6.17. Залежності інтерквартильного розмаху значень похідних h'_q від ступеня заповнення ЦЗК СД Δ_C при мінімальних $G = G_{min}$ *(а)* і максимальних $G = G_{max}$ *(б)* значеннях вагового параметра G для сильних ($q \in [0; q_{max}]$) флуктуацій значень яскравості пікселів стеганограм

значень похідної h'_q для слабких (рис. 6.16) і сильних (рис. 6.17) флуктуацій значень яскравості пікселів ЦЗ. Це обумовлено мінімальними змінами кореляційних характеристик шумів ЦЗК при вбудовуванні СД, що не виявляються при використанні СУЕГ h_q. Тому становить інтерес дослідження змін фрактальних характеристик шумів ЦЗ (ширини Δ_{α_q} і розмаху Δ_{f_q} мультифрактального спектра $f_q(\alpha_q)$), обумовлених приховуванням повідомлень до ОПЦЗК з використанням СР цифрового зображення-контейнера.

Залежності середніх значень ширини Δ_{α_q} (6.38) й розмаху Δ_{f_q} (6.39) мультифрактального спектра $f_q(\alpha_q)$ від ступеня заповнення ЦЗК стеганоданими Δ_C при мінімальному ($G = G_{min}$) і максимальному ($G = G_{max}$) значеннях вагового параметра G для слабких ($q \in [q_{min}; 0]$) і сильних ($q \in [0; q_{max}]$) флуктуацій значень яскравості пікселів стеганограм наведені на рис. 6.18.

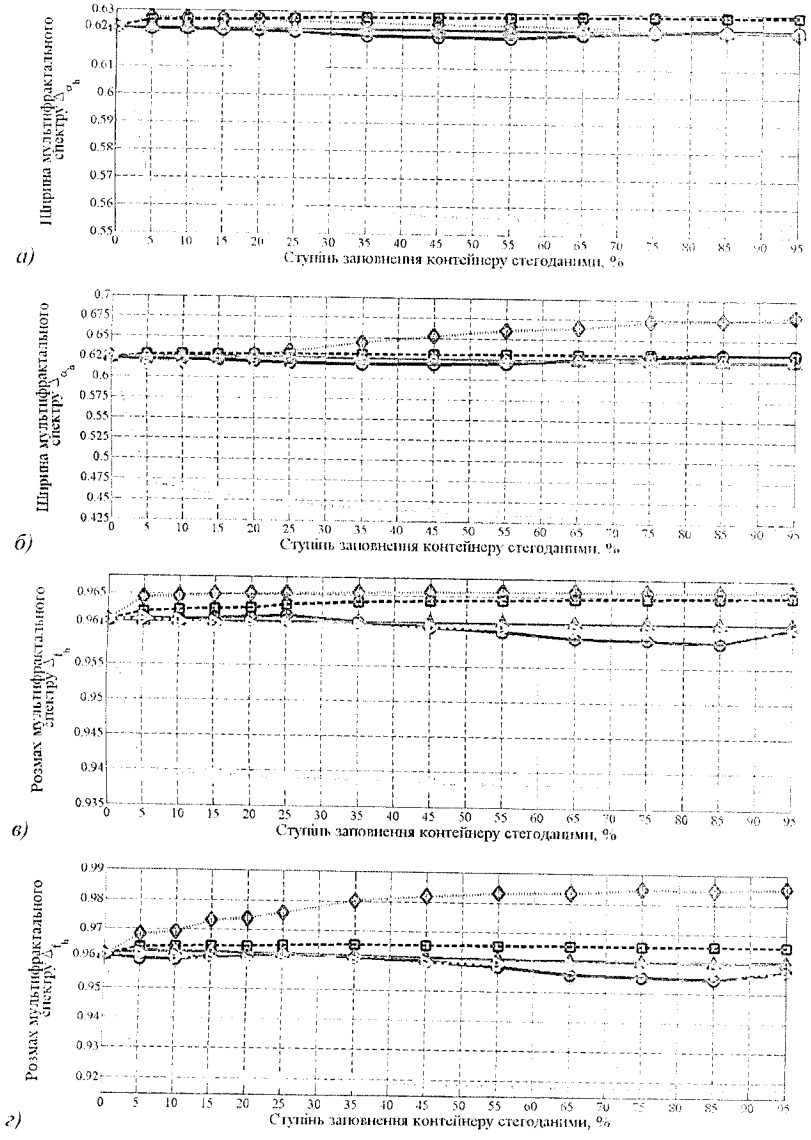


Рис. 6.18. Залежності середніх значень ширини Δ_{α_q} *(а, б)* й розмаху Δ_{f_q} *(в, г)* МФС $f_q(\alpha_q)$ стеганограм від ступеня заповнення ЦЗК СД за мінімальних *(а, в)* і максимальних *(б, г)* значень вагового параметра G

Вбудовування стеганоданих до ОПЦЗК з використанням сингулярного розкладу матриць ЦЗ згідно методів Агарвала та Джозефа призводить до стрибкоподібного зростання розмаху Δ_{f_q} МФС $f_q(\alpha_q)$ при слабкому заповненні ЦЗК СД (Δ_C= 5%) та мінімальній енергії прихованих повідомлень $G = G_{min}$ (рис. 6.18, *в*). Це свідчить про нерівномірний розподіл енергії стеганоданих між компонентами шумів ЦЗ, що дозволяє обрати ефективні методи деструкції стеганограм, наприклад, шляхом додавання до ЦЗ фрактального шуму.

Також було виявлено, що формування стеганограм згідно одноетапних методів Дея та Агарвала, багатоетапних методів Джозефа та Хана, а також комплексних методів Елахіана та Ганджела призводить до різних змін ширини Δ_{α_q} й розмаху Δ_{f_q} МФС (рис. 6.18). Це дозволяє використовувати прості порогові методи для виявлення факту приховання повідомлень до ОПЦЗК, а також оцінки кількості етапів обробки ЦЗК.

Таким чином, при проведенні ПСА стеганограм з даними, вбудованими до ОПЦЗК, можливо скоротити загальну кількість кореляційних і фрактальних характеристик шумів ЦЗ на 85% — від 182 до 27 параметрів. Для оцінки точності виявлення стеганограм, сформованих згідно одноетапних, багатоетапних і комплексних методів, при використанні скороченого переліку кореляційних і фрактальних характеристик шумів ЦЗ було проведено модифікацію стеганодетектора SD_{MFDFA}. Налаштування модифікованого стеганодетектору SD^{mod}_{MFDFA} проводилося аналогічно до SD_{VgA} з використанням ансамблю класифікаторів [161].

Порівняння ймовірності виявлення стеганограм з використанням вихідного SD_{MFDFA} і модифікованого SD^{modif}_{MFDFA} стеганодетекторів проводилася на пакеті з 2500 тестових зображень (табл. 6.2). Значення ймовірності виявлення стеганограм (метрика AUC) і середня тривалість їхньої обробки $\langle T \rangle$ при застосуванні стеганодетекторів SD_{MFDFA} і SD^{modif}_{MFDFA}, статистичного стеганодетектора SD_{CDF}, а також УСД Авсібаша для виявлення СД, вбудованих до каналу синього кольору ЦЗК згідно одноетапних, багатоетапних і комплексних методів, у випадках слабкого (AUC_{min}, Δ_C= 5%) при $G = G_{min}$ та сильного (AUC_{max}, Δ_C= 85%) при $G = G_{max}$ заповнення ЦЗК стеганоданими наведені у табл. 6.15.

Використання модифікованого стеганодетектора SD^{modif}_{MFDFA} дозволило зменшити середню тривалість обробки ЦЗ на 27% — від 9.7 до 7.1 с — без суттєвого зниження точності виявлення стеганограм. Зміни ймовірності детектування прихованих повідомлень (метрика AUC) при використанні стеганодетекторів SD_{MFDFA} і SD^{modif}_{MFDFA} не перевищують $\Delta AUC_{max} \approx 0.009$ ($\Delta AUC_{min} \leq 0.067$) у випадку сильного (слабкого) заповнення ЦЗК стеганоданими (табл. 6.15).

Таблиця 6.15

Значення AUC і середня тривалість обробки $\langle T\rangle$ зображення при застосуванні стеганодетекторів SD_{MFDFA} і SD^{modif}_{MFDFA}, статистичного стеганодетектора SD_{CDF} й УСД Авсібаша (гаусівська фільтрація стеганограм) для виявлення СД, вбудованих до каналу синього кольору ЦЗК згідно одностанних, багатостанних і комплексних методів, у випадках слабкого (AUC_{min}) й сильного (AUC_{max}) заповнення ЦЗК

		Статистична модель CDF	УСД Авсібаша	МФФА: SD_{MFDFA}	МФФА: SD^{modif}_{MFDFA}
Метод Дея	AUC_{max} / AUC_{min}	0.999 / 0.725	0.915 / 0.551	0.999 / 0.977	0.982 / 0.911
	$\langle T\rangle$, сек	11.4	5.3	9.7	7.1
Метод Агарвала	AUC_{max} / AUC_{min}	0.993 / 0.724	0.646 / 0.514	0.999 / 0.908	0.978 / 0.889
	$\langle T\rangle$, сек	11.8	5.8	10.1	7.3
Метод Джозефа	AUC_{max} / AUC_{min}	0.999 / 0.603	0.859 / 0.507	0.999 / 0.969	0.995 / 0.902
	$\langle T\rangle$, сек	11.3	5.5	9.8	6.9
Метод Хана	AUC_{max} / AUC_{min}	0.999 / 0.997	0.999 / 0.997	0.999 / 0.936	0.993 / 0.928
	$\langle T\rangle$, сек	11.2	5.2	9.5	7.0
Метод Елахіана	AUC_{max} / AUC_{min}	0.999 / 0.991	0.720 / 0.713	0.999 / 0.756	0.998 / 0.717
	$\langle T\rangle$, сек	11.0	5.3	9.6	7.1
Метод Ганджела	AUC_{max} / AUC_{min}	0.999 / 0.998	0.943 / 0.671	0.999 / 0.975	0.997 / 0.943
	$\langle T\rangle$, сек	11.1	5.2	9.5	7.1

Враховуючи високу ймовірність виявлення стеганограм з використанням стеганодетектора SD^{modif}_{MFDFA} в широкому діапазоні значень рівня заповнення ЦЗК стеганоданими ($\Delta_C \in [10; 85]$) та енергії прихованих повідомлень (коефіцієнт G, табл. 6.1), даний стеганодетектор можна рекомендувати до використання в якості стандартного при проведенні ПСА цифрових зображень.

Зниження ймовірності виявлення стеганограм з даними, вбудованими до ОПЦЗК, у випадку слабкого заповнення ЦЗК СД ($\Delta_C \leq 10\%$) та мінімальній енергії СД ($G = G_{min}$) при використанні стеганодетектора SD^{modif}_{MFDFA} (табл. 6.15) обумовлено мінімальними змінами кореляційних і фрактальних характеристик ЦЗК, що не можуть бути виявленими при використанні МФФА. Тому перспективним є застосування спеціальних методів структурного аналізу, зокрема мультифрактального, для підвищення ймовірності виявлення стеганограм у найбільш складних випадках ПСА ЦЗ.

6.7.2.2. *Мультифрактальний аналіз стеганограм*

За результатами дослідження фрактальних характеристик ЦЗК і стеганограм, сформованих згідно одноетапних, багатоетапних і комплексних методів, було встановлено, що найбільших змін при вбудовуванні СД зазнають граничні фрактальні розмірності $D_{-\infty}(D_{+\infty})$, а також ширина Δ_α.

$$\Delta_\alpha = \alpha_{max} - \alpha_{min},$$

і розмах Δ_f

$$\Delta_f = f_{max}(\alpha) - f_{min}(\alpha)$$

мультифрактального спектра $f(\alpha)$ [260].

Залежності значень граничних фрактальних розмірностей $D_{-\infty}(D_{+\infty})$ стеганограм з даними, вбудованими до ОПЦЗК згідно одноетапних, багатоетапних і комплексних стеганометодів, від ступеня заповнення ЦЗК стеганоданими Δ_C при мінімальному ($G = G_{min}$) і максимальному ($G = G_{max}$) значенні вагового параметра G наведені на рис. 6.19, 6.20.

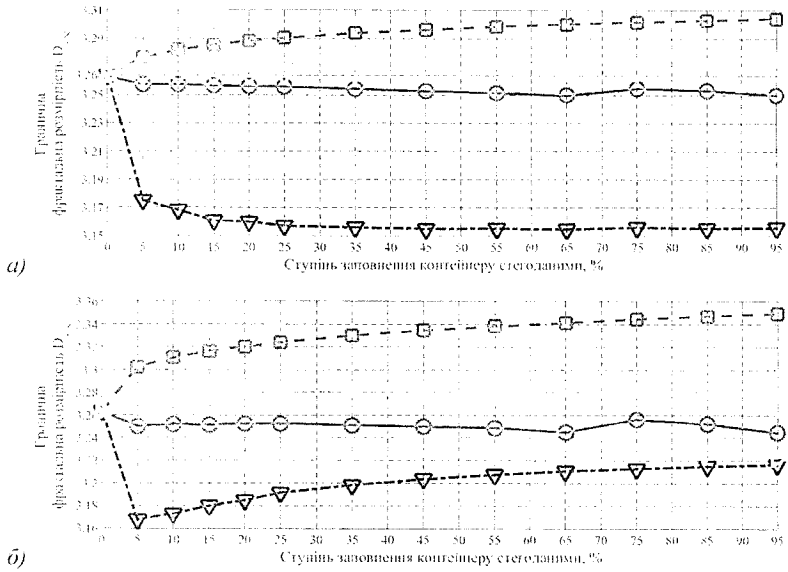


Рис. 6.19. Залежності значень граничної фрактальної розмірності $D_{-\infty}$ для стеганограм з даними, вбудованими до ОПЦЗК, від ступеня заповнення ЦЗК СД при мінімальних *(а)* і максимальних *(б)* значеннях вагового параметра G

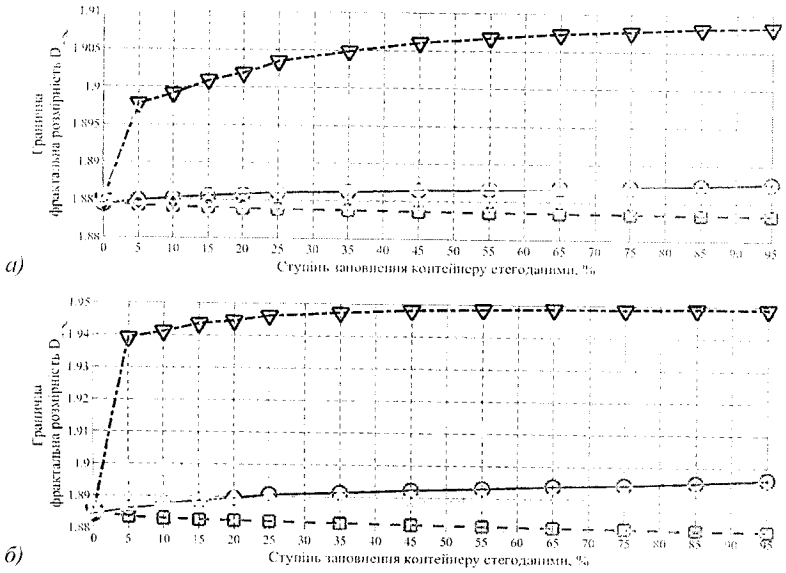


Рис. 6.20. Залежності значень граничної фрактальної розмірності $D_{+\infty}$ для стеганограм з даними, вбудованими до ОПЦЗК, від ступеня заповнення ЦЗК СД при мінімальних *(а)* і максимальних *(б)* значеннях вагового параметра G

Демаскуючою ознакою стеганограм, сформованих згідно методу Агарвала, є суттєве зростання граничної фрактальної розмірності $D_{-\infty}$ (рис. 6.19) ЦЗК навіть у випадку слабкого його заповнення стегоданими ($\Delta_C = 5\%$) та мінімальній енергії останніх $G = G_{min}$.

Різний характер змін граничних фрактальних розмірностей $D_{-\infty}$ ($D_{+\infty}$) стеганограм при використанні спектральних (двовимірного ДВП) і спеціальних перетворень (сингулярного розкладу) ЦЗК для приховання повідомлень згідно методів Дея та Хана, а також Агарвала та Джозефа (рис. 6.19, 6.20) дає можливість не лише виявляти факт вбудовування СД, але й визначати тип перетворень графічного контейнера, використаних при формуванні стеганограми.

Відносно малі зміни граничних фрактальних розмірностей $D_{-\infty}(D_{+\infty})$ для стеганограм, сформованих згідно методу Елахіана (рис. 6.19, 6.20), суттєво ускладнюють їхнє виявлення з використанням лише СУФР D_q. Тому становить інтерес використання параметрів МФС $f(\alpha)$ — ширини Δ_α й розмаху Δ_f — для підвищення ймовірності виявлення вбудованих до ОПЦЗК згідно зазначеного методу СД.

Залежності ширини Δ_α й розмаху Δ_f МФС $f(\alpha)$ стеганограм з даними, вбудованими до ОПЦЗК згідно одноетапних методів Дея та Агарвала, багатоетапних методів Джозефа та Хана, а також комплексних методів Елахіана та Ганджела, від ступеня заповнення ЦЗК стеганоданими Δ_C і значення вагового параметра G наведені на рис. 6.21.

Як видно з рис. 6.21,*г*, *к*, характер розмаху Δ_f МФС $f(\alpha)$ стеганограм, сформованих згідно багатоетапного методу Хана, суттєво залежить від значення вагового параметра G: зменшення G ($G \to G_{min}$) призводить до зростання Δ_f (збільшення нерівномірності розподілу енергії між окремими монофрактальними компонентами ЦЗК), а при $G \to G_{max}$ — до наближення стеганограми до монофрактального сигналу ($\Delta_f \to 0$). Це дає можливість використовувати прості порогові методи для виявлення стеганограм, а також для оцінки енергії стеганоданих.

Формування стеганограм згідно методу Елахіана призводить до стрибкоподібного збільшення ширини Δ_f МФС (рис. 6.21,*д*), що відповідає появі нових монофрактальних компонент ЦЗК, а не перерозподілу енергії між існуючими (наявними) компонентами. Отриманий результат є неочікуваним і може бути поясненим використанням перетворення Арнольда (6.8) для обробки приховуваних повідомлень з метою наближення їхніх характеристик до псевдовипадкового фрактального сигналу [261]. Це призводить до появи нових монофрактальних компонент ЦЗК при вбудовуванні модифікованих СД до ОПЦЗК і дозволяє використовувати методи порогової обробки ширини Δ_f МФС $f(\alpha)$ для виявлення сформованих згідно методу Елахіана стеганограм.

Отже, використання лише граничних фрактальних розмірностей $D_{-\infty}$ ($D_{+\infty}$), а також ширини Δ_α й розмаху Δ_f МФС $f(\alpha)$ дозволяє скоротити кількість використовуваних при проведенні ПСА ЦЗ фрактальних характеристик на 29% — з 14 до 10.

Для оцінки ймовірності виявлення вбудованих до ОПЦЗК стеганограм з використанням первинного й скороченого переліків фрактальних характеристик ЦЗ стеганодетектор SD_{MFA} модифіковано до виду SD_{MFA}^{modif}, налаштування якого проводилося аналогічно до статистичних стеганодетекторів SD_{SPAM}, $SD_{CC\text{-}PEV}$ і SD_{CDF} з використанням АК.

Порівняння точності виявлення стеганограм при використанні вихідного SD_{MFA} й модифікованого SD_{MFA}^{modif} стеганодетекторів проводилася на тестовому пакеті з 2500 ЦЗ (табл. 6.2). Значення ймовірності виявлення (метрика AUC) і середній час обробки $\langle T \rangle$ стеганограм при застосуванні стеганодетекторів SD_{MFA} і SD_{MFA}^{modif}, статистичного стеганодетектора SD_{CDF} й УСД Авсібаша для виявлення СД, вбудованих до каналу синього кольору ЦЗК згідно одноетапних, багатоетапних і комплексних методів, у випадках слабкого (AUC_{min}, $\Delta_C = 5\%$) при $G = G_{min}$ і сильного (AUC_{max}, $\Delta_C = 85\%$) при $G = G_{max}$ заповнення ЦЗК СД наведені у табл. 6.16.

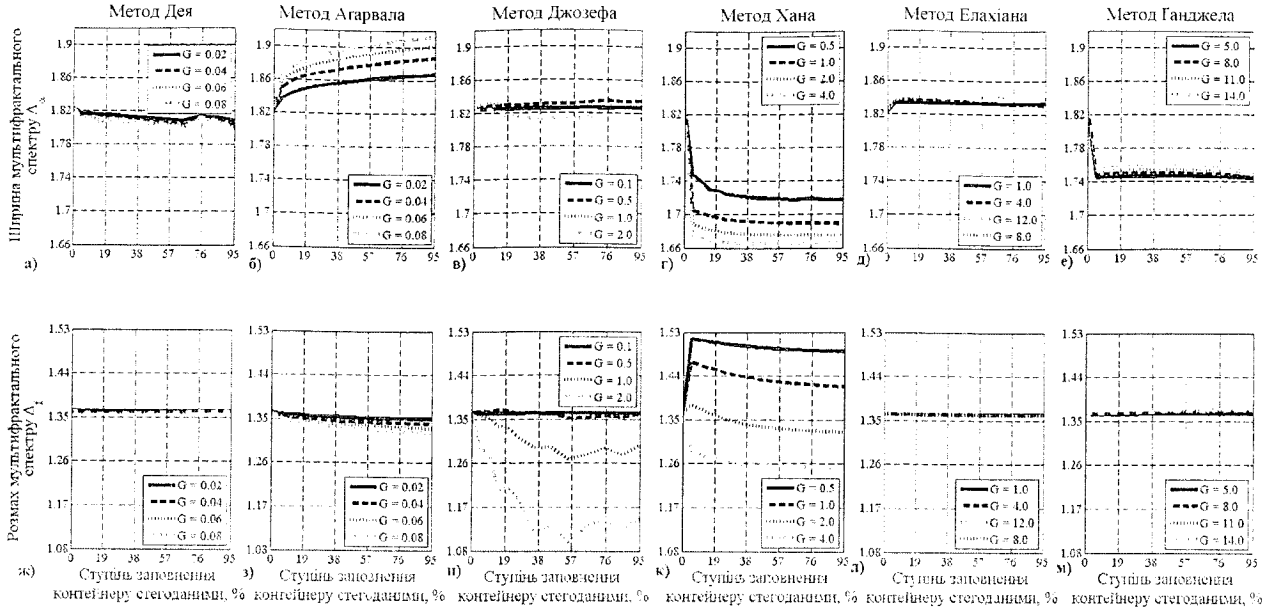


Рис. 6.21. Залежності ширини Δ_α *(а-е)* й розмаху Δ_f *(ж-м)* мультифрактального спектра $f(\alpha)$ від ступеня заповнення ЦЗК стеганограмами й значення вагового параметра G стеганограм, сформованих згідно: *(а, ж)* – методу Дея; *(б, з)* – методу Агарвала; *(в, и)* – методу Джозефа; *(г, к)* – методу Хана; *(д, л)* – методу Елахіана; *(е, м)* – методу Ганджела

Таблиця 6.16

Значення AUC і середня тривалість обробки $\langle T \rangle$ зображення при застосуванні стеганодетекторів SD_{MFA} і SD_{MFA}^{modif}, статистичного стеганодетектора SD_{CDF} й УСД Авсібаша для виявлення СД, вбудованих до каналу синього кольору ЦЗК згідно одноетапних, багатоетапних і комплексних методів, у випадках слабкого (AUC_{min}) при $G = G_{min}$ і сильного (AUC_{max}) при $G = G_{max}$ заповнення ЦЗК

		Статистична модель CDF	УСД Авсібаша	МФА: SD_{MFA}	МФА: SD_{MFA}^{modif}
Метод Дея	AUC_{max} / AUC_{min}	0.999 / 0.725	0.915 / 0.551	0.999 / 0.986	0.981 / 0.941
	$\langle T \rangle$, сек	11.4	5.3	7.7	7.4
Метод Агарвала	AUC_{max} / AUC_{min}	0.993 / 0.724	0.646 / 0.514	0.991 / 0.971	0.979 / 0.964
	$\langle T \rangle$, сек	11.8	5.8	7.5	7.4
Метод Джозефа	AUC_{max} / AUC_{min}	0.999 / 0.603	0.859 / 0.507	0.999 / 0.962	0.985 / 0.959
	$\langle T \rangle$, сек	11.3	5.5	7.6	7.2
Метод Хана	AUC_{max} / AUC_{min}	0.999 / 0.997	0.999 / 0.997	0.999 / 0.955	0.991 / 0.950
	$\langle T \rangle$, сек	11.2	5.2	7.5	7.3
Метод Елахіана	AUC_{max} / AUC_{min}	0.999 / 0.991	0.720 / 0.713	0.970 / 0.950	0.963 / 0.942
	$\langle T \rangle$, сек	11.0	5.3	7.4	7.2
Метод Ганджела	AUC_{max} / AUC_{min}	0.999 / 0.998	0.943 / 0.671	0.999 / 0.980	0.994 / 0.976
	$\langle T \rangle$, сек	11.1	5.2	7.5	7.2

Використання модифікованого стеганодетектора SD_{MFA}^{modif} дозволило досягти співставної ймовірності виявлення стеганограм (метрика AUC) з випадком використання первинного стеганодетектора SD_{MFA} — зміни значення метрики AUC не перевищують $\Delta AUC_{min} \leq 0.045$ у випадку слабкого заповнення ЦЗК стеганоданими, та $\Delta AUC_{max} \leq 0.018$ у разі сильного заповнення (табл. 6.16). Скорочення переліку фрактальних характеристик ЦЗ на 29% (з 14 до 10) дозволило зменшити середню тривалість обробки ЦЗ з використанням модифікованого стеганодетектора SD_{MFA}^{modif} на 4% — з 7,5 до 7,2 с.

Таким чином, можна рекомендувати використовувати модифікований стеганодетектор SD_{MFA}^{modif} для виявлення стеганограм у найбільш складних випадках ПСА — при слабкому заповненні ЦЗК стеганоданими мінімальної енергії ($\Delta_C \leq 10\%$, $G = G_{min}$).

За результатами проведених досліджень змін кореляційних і фрактальних характеристик стеганограм та їхніх шумових компонент з

використанням методів варіограмного (див. п. 6.7.1), мультифрактального флуктуаційного (підп. 6.7.2.1) й мультифрактального (підп. 6.7.2.2) аналізів, були сформовані кластери ДМО для одноетапних, багатоетапних і комплексних СМ. У табл. 6.17 наведено кількості кореляційних і фрактальних параметрів для первинного й модифікованого просторів ознак стеганодетектора на основі методів структурного аналізу ЦЗ, а також відповідні простору ознак розмірності для статистичних стеганодетекторів SD_{SPAM}, $SD_{CC\text{-}PEV}$, SD_{CDF} та УСД Авсібаша.

Таблиця 6.17

Кількість елементів простору ознак стеганодетектора, налаштованого з використанням методів структурного й статистичного аналізу зображень, а також універсального стеганодетектора Авсібаша

Тип використовуваного простору ознак	Кількість елементів у первинному/модифікованому просторові ознак
Кореляційні характеристик зображень — ВА	39/5
Кореляційні й фрактальні характеристики зображень — МФФА	182/27
Фрактальні характеристик зображень — МФА	14/10
Статистична модель *SPAM*	686/-
Статистична модель *CC-PEV*	548/-
Статистична модель *CDF*	1234/-
Метрики якості ЦЗ (УСД Авсібаша)	14/-

Формування кластерів ДМО дало можливість скоротити загальну кількість кореляційних і фрактальних характеристик зображень, що використовуються при проведенні структурного стеганоаналізу ЦЗ, на 82% — з 39+182+14=235 до 5+27+10=42 (табл. 6.17).

Зважаючи на суттєве скорочення потужності кластера ДМО для стеганограм, сформованих згідно одноетапних, багатоетапних і комплексних СМ, актуальною є задача порівняння впливу використовуваного простору ознак структурних стеганодетекторів SD_{VgA}^{modif}, SD_{MFDFA}^{modif} і SD_{MFA}^{modif}, статистичного стеганодетектора SD_{CDF} й УСД Авсібаша $SD_{Avcibaş}$.

Для дослідження впливу використовуваного простору ознак (метрика *INs*) на точність виявлення стеганограм з даними, вбудованими до ОПЦЗК, при використанні методів структурного й статистичного стеганоаналізу, а також УСД Авсібаша було проведено тестування модифікованих стеганодетекторів SD_{VgA}^{modif}, SD_{MFDFA}^{modif} і SD_{MFA}^{modif}, а також SD_{CDF} та $SD_{Avcibaş}$ на тестовому пакеті з 2500 цифрових зображень при використанні стеганографічних даних типу «Портрет» (табл. 6.2).

Значення метрики INs при використанні СМ структурного й статистичного стеганоаналізу, а також УСД Авсібаша для виявлення стеганограм з даними, вбудованими до ОПЦЗК, у випадках слабкого (INs_{min}, $\Delta_C = 5\%$) при $G = G_{min}$ і сильного (INs_{max}, $\Delta_C = 85\%$) при $G = G_{max}$ заповнення ЦЗК стеганоданими наведені у табл. 6.18.

Таблиця 6.18

Значення INs при використанні методів структурного й статистичного стеганоаналізу, а також УСД Авсібаша для виявлення стеганограм з даними, вбудованими до ОПЦЗК, у випадках слабкого (INs_{min}) і сильного (INs_{max}) заповнення ЦЗК

		SD_{CDF}	$SD_{Avcibaş}$	SD_{VgA}^{modif}	SD_{MFDFA}^{modif}	SD_{MFA}^{modif}
Метод Дея	INs_{max} / INs_{min}	0.986 / 0.582	0.829 / 0.108	0.999 / 0.283	0.999 / 0.854	0.996 / 0.874
Метод Агарвала	INs_{max} / INs_{min}	0.917 / 0.312	0.305 / 0.026	0.999 / 0.198	0.999 / 0.680	0.941 / 0.782
Метод Джозефа	INs_{max} / INs_{min}	0.998 / 0.144	0.721 / 0.005	0.999 / 0.472	0.999 / 0.841	0.994 / 0.808
Метод Хана	INs_{max} / INs_{min}	0.999 / 0.987	0.999 / 0.994	0.999 / 0.256	0.999 / 0.734	0.988 / 0.860
Метод Елахіана	INs_{max} / INs_{min}	0.993 / 0.982	0.459 / 0.437	0.999 / 0.195	0.999 / 0.475	0.893 / 0.817
Метод Ганджела	INs_{max} / INs_{min}	0.999 / 0.997	0.891 / 0.342	0.999 / 0.167	0.999 / 0.800	0.968 / 0.895

Виключення кореляційних і фрактальних характеристик зображень-контейнерів, що при вбудовуванні стеганографічних даних до ОПЦЗК змінюються несуттєво, дало можливість досягти високих значень метрики INs для стеганодетекторів SD_{VgA}^{modif}, SD_{MFDFA}^{modif} і SD_{MFA}^{modif} у випадку як сильного ($\Delta_C = 85\%$), так і слабкого ($\Delta_C = 5\%$) заповнення ЦЗК (див. табл. 6.18).

У випадку формування стеганограм згідно одноетапних методів Дея та Агарвала, а також двоетапного методу Джозефа використання сформованих кластерів ДМО відносно малої потужності ($d_{VgA}^{modif} = 5$, $d_{MFDFA}^{modif} = 27$, $d_{MFA}^{modif} = 10$) дає можливість суттєво підвищити значення метрики INs (табл. 6.18) у порівнянні з випадком використання статистичних моделей ЦЗ високої розмірності ($d_{CDF} = 1234$, табл. 6.17), а також дозволяє значно скоротити час налаштування стеганодетекторів і пом'якшити вимоги щодо об'єму пакета тестових цифрових зображень, що є вагомою перевагою методів структурного аналізу стеганограм.

6.8. Програмний комплекс пасивного стеганоаналізу цифрових зображень

За результатами дослідження змін кореляційних і фрактальних характеристик ЦЗК, обумовлених вбудовуванням стеганографічних даних згідно одноетапних, багатоетапних і комплексних стеганографічних методів, нами були сформовані кластери ДМО для кожного СМ. Відносно мала потужність даних кластерів ознак (табл. 6.17) стеганограм дає можливість скоротити час налаштування стеганодетекторів і знизити вимоги щодо об'єму пакета тестових зображень, забезпечуючи разом з тим високу ймовірність розпізнання стеганограм з даними, вбудованими до ОПЗК.

При формуванні стеганограм використання спектральних (двовимірне ДВП) та спеціальних (сингулярний розклад) перетворень матриць цифрових зображень-контейнерів призводить до різних змін кореляційних і фрактальних характеристик останніх. Урахування характеру цих змін при проведенні ПСА ЦЗ дає можливість не лише виявляти факт приховання повідомлень, а й визначати особливості використаного стеганографічного методу — зокрема, тип перетворення ЦЗК, кількість етапів обробки графічного контейнера та стеганоданих, енергію прихованих повідомлень. Це дозволяє розробити ефективні методи деструкції вбудованих стеганоданих при забезпеченні мінімальних візуальних змін зображень-контейнерів. Тому актуальною є задача розробки загального стеганодетектора (ЗСД) для цифрових зображень, що дозволить не тільки виявляти факт приховання повідомлень до їхньої області перетворення, але й надавати рекомендації щодо деструкції стеганоданих.

Інтеграція розроблених структурних стеганодетекторів SD^{mod}_{VgA}, SD^{mod}_{MFDFA} і SD^{mod}_{MFA}, а також ЗСД до складу єдиного програмного комплексу дозволить автоматизувати процес виявлення стеганограм з даними, вбудованими до ОПЦЗК з використанням спектральних (двовимірні ДВП і ДКП) та спеціальних перетворень (сингулярний розклад) матриць яскравості пікселів графічного контейнера. Розробкою програмних комплексів ПСА цифрових зображень займаються провідні іноземні *IT*-компанії, зокрема корпорація *Cisco* [262, 263]. Проте використання даних комплексів для забезпечення інформаційної безпеки об'єктів критичної інфраструктури України — наприклад, систем комунікації, систем керування виробництвом небезпечних матеріалів і компонентів тощо — є неможливим через правові обмеження щодо продажу та використання програмних комплексів державам, що не є учасниками військово-політичного альянсу *NATO* (*North Atlantic Treaty Organization*). Внаслідок цього, розроблений комплекс прикладних програм для ПСА ЦЗ може бути використаним в якості складової частини вітчизняних систем моніторингу й контролю інформаційно-комунікаційних систем.

6.8.1. Загальний стеганодетектор цифрових зображень

6.8.1.1. Розробка загального стеганодетектора

Значна кількість відомих методів ПСА ЦЗ заснована на використанні двокласових (бінарних) стеганодетекторів, що відносять ЦЗ до класу пустих зображень-контейнерів або стеганограм шляхом порівняння параметрів досліджуваного зображення з апріорно відомими характеристиками зазначених класів. У випадку наявності апріорних даних щодо особливостей K ($K > 2$) стеганографічних методів (наприклад, кількості етапів обробки ЦЗК і стеганоданих) пасивний стеганоаналіз ЦЗ може бути представлений як задача віднесення досліджуваного зображення до одного з $K + 1$ класів — зображення-контейнера або стеганограми, сформованих згідно відомих СМ. Розв'язання такої задачі вимагає переходу від бінарного до узагальненого (багатокласового) стеганодетектора.

Зважаючи на необхідність використання обчислювально складних методів класифікації для побудови відокремлювальних гіперплощин між класами, налаштування загального стеганодетектора є нетривіальною задачею [184, 192]. Для налаштування ЗСД нами був використаний *One-Against-All* (*OVA*) метод [184, 264, 265], що полягає у послідовному налаштуванні «базових» класифікаторів для виявлення лише окремих класів (методів формування стеганограм) з використанням методів структурного аналізу ЦЗ. Структурна схема системи налаштування загального структурного стеганодетектора наведена на рис. 6.22.

Обробка пакета тестових зображень з використанням загального структурного стеганодетектора проводиться в декілька етапів [264, 265] (див. рис. 6.22). На першому етапі на вхід блока №1 передаються тестові цифрові зображення, стеганодані й параметри налаштування стеганокодера (метод приховання повідомлень, ступінь заповнення ЦЗК СД, значення вагового параметра G) для формування стеганограм.

Пусті зображення-контейнери та сформовані на їх основі стеганограми передаються до блока №2, в якому визначаються кореляційні й фрактальні характеристики цих зображень з використанням розроблених алгоритмів структурного аналізу [161] — варіограмного, флуктуаційного та мультифрактального.

Розраховані кореляційні й фрактальні характеристики ЦЗК і стеганограм подаються на вхід блока №3, де використовуються для налаштування базового класифікатора — віднесення аналізованих зображень до класу пустих контейнерів або стеганограм. Налаштування базового класифікатора ЗСД і модуля класифікаторів для СМ блока №3 проводилося з використанням АК аналогічно до структурних стеганодетекторів SD^{mod}_{VgA}, SD^{mod}_{MFDFA} і SD^{mod}_{MFA}.

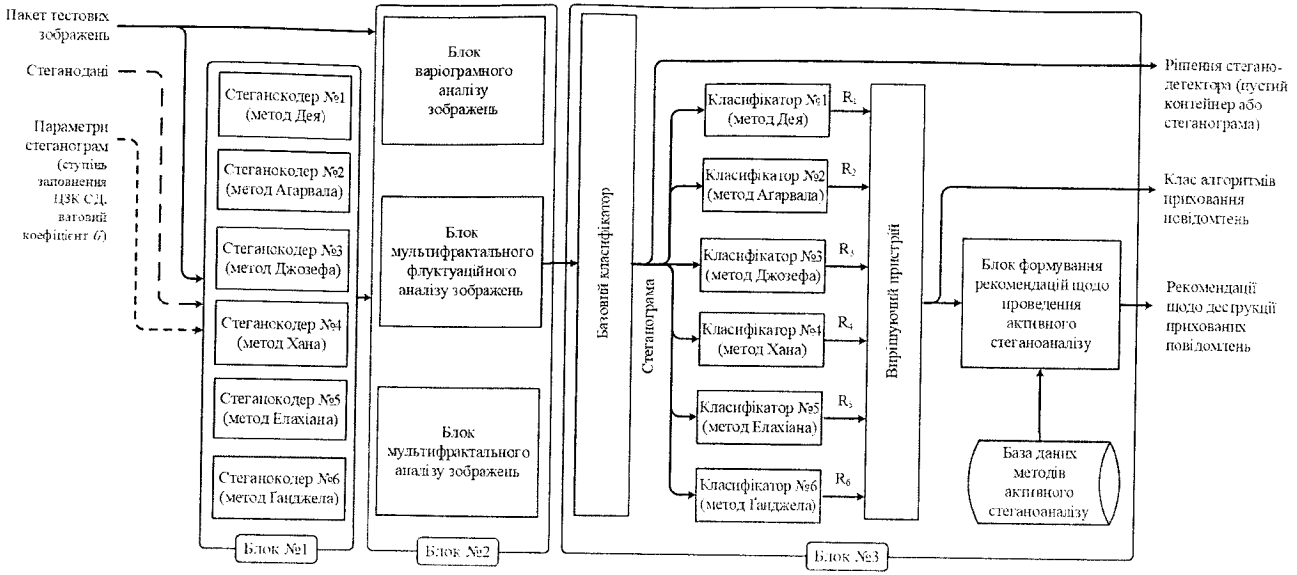


Рис. 6.22. Структурна схема системи налаштування загального структурного стеганодетектора цифрових зображень згідно методу *One-Against-All (OVA)*

У випадку класифікації досліджуваного ЦЗ як пустого ЦЗК, відповідне рішення передається на вихід ЗСД. У протилежному випадку, кореляційні й фрактальні характеристики зображень, віднесених базовим класифікатором до класу стеганограм, передаються на вхід класифікаторів №№1–6, що використовуються для виявлення стеганограм, сформованих згідно одноетапних методів Дея та Агарвала (класифікатори №1 і №2), багатоетапних методів Джозефа та Хана (№3 і №4) та комплексних методів Елахіана та Ганджела (№5 і №6).

Результати роботи класифікаторів — імовірність R_j, $j \in [1;6]$ віднесення зображення до j-го класу (методу вбудовування стеганоданих до ОПЦЗК) — передаються до вирішуючого пристрою. Рішення щодо віднесення зображення до класу стеганограм, сформованих згідно одноетапних, багатоетапних чи комплексних методів, приймається шляхом вибору найбільшого значення ймовірності — R_{max}.

Згідно рішення ЗСД щодо віднесення досліджуваного ЦЗ до класів стеганограм додатково формується перелік рекомендацій щодо налаштування методів активного стеганоаналізу для забезпечення деструкції прихованих повідомлень при фіксованому рівні візуальних змін графічного контейнера [266–269].

6.8.1.2. Виявлення стеганограм з використанням загального стеганодетектора

Для оцінки ймовірності віднесення зображення до класів пустих ЦЗК чи стеганограм було проведено налаштування й тестування ЗСД з використанням псевдовипадкової вибірки без повторень 770 зображень з тестового пакета (табл. 6.2). Сформована вибірка була псевдовипадковим чином розділена на навчальний і контрольний пакети, що містили в собі зображення-контейнери та стеганограми, сформовані згідно одноетапних, багатоетапних і комплексних методів (700 «навчальних» та 70 «тестових» зображень).

Налаштування ЗСД проводилося з використанням системи, наведеної у підп. 6.8.1.1. Для визначення параметрів окремих класифікаторів ЗСД на вхід системи виконувалася послідовна подача зображень, що належать кожному з сімох класів: пусті ЦЗК (клас №0) та стеганограми, сформовані згідно одноетапних методів Дея (клас №1) та Агарвала (клас №2), багатоетапних методів Джозефа (клас №3) та Хана (клас №4), а також комплексних методів Елахіана (клас №5) та Ганджела (клас №6). При тестуванні налаштованого ЗСД на вхід блока №2 (рис. 6.22) подавалася псевдовипадковим чином перемішана вибірка пустих ЦЗК з контрольного пакета й стеганограм, сформованих згідно одноетапних, багатоетапних та комплексних методів.

Значення міжкласової помилки — ймовірності віднесення зображення, що належить i-му класу, до j-го класу $(i \neq j)$ — для випадку слабкого $(\Delta_C = 10\%,)$ при $G = G_{min}$ та сильного $(\Delta_C = 85\%)$ при $G = G_{max}$ заповнення зображення-контейнера стеганографічними даними та прихованні СД типу «Портрет» до каналу синього кольору ЦЗК наведені у табл. 6.19.

Таблиця 6.19

Імовірності віднесення ЗСД ЦЗ до класів пустих контейнерів (№0) і стеганограм (№1–6) у випадку слабкого ($\Delta_C = 10\%$) при $G = G_{min}$ та сильного ($\Delta_C = 85\%$) при $G = G_{max}$ заповнення стеганоданими

Істинні мітки класів / Рішення ЗСД	№0	№1	№2	№3	№4	№5	№6
$\Delta_C = 10\%, G = G_{min}$							
№0	**0.935**	0.055	0.001	2.77e-4	4.85e-5	3.18e-6	1.33e-7
№1	0.053	**0.883**	0.061	0.003	5.34e-4	9.63e-5	0.99e-6
№2	0.002	0.060	**0.860**	0.074	0.004	6.90e-4	6.36e-6
№3	3.54e-4	0.003	0.073	**0.853**	0.067	0.004	2.52e-4
№4	2.21e-5	5.35e-4	0.004	0.076	**0.859**	0.058	0.002
№5	1.34e-6	3.26e-5	6.74e-4	0.003	0.057	**0.912**	0.028
№6	0.93e-6	6.96e-6	2.39e-6	1.85e-4	0.002	0.028	**0.970**
$\Delta_C = 85\%, G = G_{max}$							
№0	**0.998**	0.001	0.001	2.77e-4	5.33e-5	4.41e-6	1.12e-7
№1	0.001	**0.995**	0.002	0.003	1.11e-4	7.12e-5	2.63e-6
№2	1.13e-4	0.002	**0.992**	0.002	0.004	4.89e-4	5.38e-6
№3	1.94e-5	0.002	0.001	**0.991**	0.005	0.001	1.74e-4
№4	9.82e-6	1.12e-4	0.001	0.001	**0.996**	0.001	0.001
№5	6.85e-6	1.63e-5	8.29e-4	0.002	0.001	**0.994**	0.003
№6	0.54e-6	5.47e-6	1.45e-6	1.58e-4	0.001	0.002	**0.997**

Як видно з табл. 6.19, використання ЗСД дозволяє з високою ймовірністю визначати приналежність ЦЗ як до класу пустого ЦЗК (клас №0), так і до класів стеганограм, сформованих згідно одноетапних методів Дея та Агарвала (класи №1 і №2 відповідно), багатоетапних методів Джозефа та Хана (класи №3 і №4 відповідно) та комплексних методів Елахіана та Ганджела (класи №5 і №6 відповідно) у випадку як сильного, так і слабкого заповнення ЦЗК стеганоданими.

Також був розглянутий випадок обробки ЗСД стеганограм, сформованих згідно методу, апріорно невідомого на етапі налаштування загального структурного стеганодетектора, — адаптивного методу *UNIWARD*

[158]. Останній заснований на вбудовуванні СД до ОПЦЗК шляхом зміни значень яскравості пікселів ЦЗК та мінімізації цільової функції $D(\ldots)$ — оцінки відмінностей між пустим зображенням-контейнером $\boldsymbol{I}_{x,y}$ та стеганограмою $\boldsymbol{S}_{x,y}$ розмірами $M \times N$ пікселів [158]:

$$D\left(\boldsymbol{I}_{x,y}, \boldsymbol{S}_{x,y}\right) = \sum_{k\in\{h,v,d\}} \sum_{u=1}^{M/2} \sum_{v=1}^{N/2} \frac{\left|W_{u,v}^{(k)}\left(\boldsymbol{I}_{x,y}\right) - W_{u,v}^{(k)}\left(\boldsymbol{S}_{x,y}\right)\right|}{W_{u,v}^{(k)}\left(\boldsymbol{I}_{x,y}\right) + \theta_{st}}, \qquad (6.47)$$

де $W_{u,v}^{(k)}\left(\boldsymbol{I}_{x,y}\right)$, $W_{u,v}^{(k)}\left(\boldsymbol{S}_{x,y}\right)$ — відповідно, деталізуючі коефіцієнти одно-рівневого двовимірного ДВП пустого ЦЗК і стеганограми; θ_{st} $(\theta_{st} > 0)$ — константа, що використовується для забезпечення стійкості розрахунку значення цільової функції (6.47) у випадку малих значень коефіцієнтів двовимірного ДВП ЦЗК $(W_{u,v}^{(k)}\left(\boldsymbol{I}_{x,y}\right) \cong 0,\ k \in \{h, v, d\})$; k $(k \in \{h, v, d\})$ — вид деталізуючих коефіцієнтів двовимірного ДВП ЦЗК, що відповідають обробці рядків $(k = h)$, стовпчиків $(k = v)$ або рядків і стовпчиків $(k = d)$ з використанням фільтра високих частот (базисного вейвлета).

При оцінці величини міжкласової помилки з використанням ЗСЛ ступінь заповнення ЦЗК стеганоданими дорівнювала $\Delta_C = 20\%$. В якості СД було використане ЦЗ типу «Портрет» (табл. 6.2), а в якості базисної функції двовимірного ДВП — вейвлет Гаара та відповідна йому скейлінг-функція. Для забезпечення найбільшої робастності отримуваних стеганограм до сучасних методів статистичного стеганоаналізу ЦЗ, значення константи θ_{st} було обрано рівним $\theta_{st} = 1$ [270].

За результатами обробки з використанням ЗСД стеганограм, сформованих згідно алгоритму *UNIWARD*, були отримані ймовірності віднесення стеганограми до кожного з класів — табл. 6.20.

Таблиця 6.20

Імовірності P_i віднесення сформованої згідно *UNIWARD* $(\Delta_C = 20\%)$ стеганограми, до i-го класу при використанні загального структурного стеганодетектора

	Клас №0	Клас №1	Клас №2	Клас №3	Клас №4	Клас №5	Клас №6
P_i	0.0825	0.0682	0.0743	**0.9899**	0.0066	0.3818	0.0035

Застосування ЗСД дає можливість не тільки з високою ймовірністю детектувати наявність у ЦЗ стеганоданих (табл. 6.20), а й встановити, що приховання повідомлень проводилося з використанням двовимірного ДВП і застосуванням додаткових перетворень стеганографічних даних для мінімізації спотворень характеристик зображення-контейнера (метод Джозефа, клас №3).

За результатами роботи ЗСД були сформовані рекомендації щодо деструкції стеганограм, сформованих згідно даного методу — жорсткої порогової обробки коефіцієнтів двовимірного ДВП за алгоритмом Бірґе-Массарта [208, 271].

Результати деструкції стеганограм при використанні як поширених методів АСА — медіанний фільтр та фільтр Ґауса ($\sigma^2 = 0.5$) розміром 3×3 пікселів, — так і рекомендованого методу (вейвлет-стиснення згідно алгоритму Бірґе-Массарта при використанні в якості базисної функції трирівневого двовимірного ДВП вейвлета Гаара та відповідної йому скейлінґ-функції) наведені на рис. 6.23.

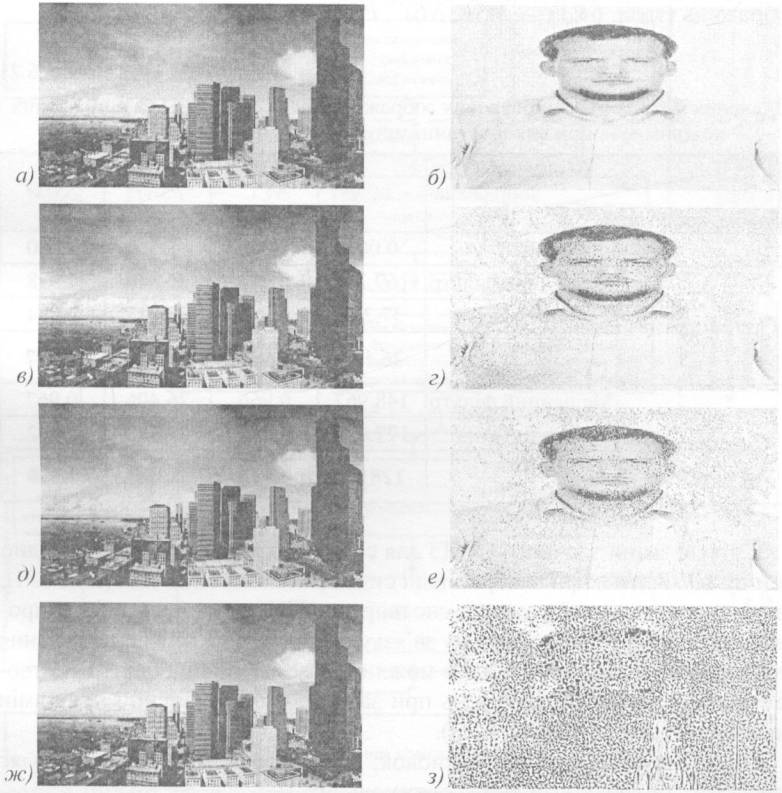


Рис. 6.23. Первинний вигляд пустого ЦЗК *(а)* і СД *(б)*; результати обробки стеганограм та вилучені стеганодані при використанні: *(в, г)* – медіанної фільтрації; *(д, е)* – ґаусівської фільтрації; *(ж, з)* – вейвлет-стиснення ЦЗ згідно алгоритму Бірґе-Массарта

За результатами порівняння первинного вигляду ЦЗК (рис. 6.23, *а*) та стеганоданих (рис. 6.23, *б*), результатів застосування поширених методів АСА (рис. 6.23, *в–е*) та рекомендованого методу деструкції стеганограми (рис. 6.23, *ж–з*), сформованої згідно методу *UNIWARD*, можна зробити висновок, що використання ЗСД дозволяє визначити оптимальний метод деструкції стеганограм — максимізувати ступінь спотворення стеганографічних даних при мінімальних візуальних змінах зображення-контейнера.

Враховуючи неможливість візуального розрізнення результатів обробки пустого ЦЗК і стеганограми, для кількісної оцінки ступеня спотворення ЦЗК і СД були розраховані стандартні МЯ цифрових зображень (табл. 6.21) — *MSE*, *NCC*, *PSNR* та *SSIM* [161].

Таблиця 6.21

Значення метрик якості цифрових зображень для стеганограм та прихованих повідомлень при використанні методів активного стеганоаналізу

Метрики якості зображень / Методи обробки зображень		*MSE*	*NCC*	*PSNR*	*SSIM*
Без обробки		0.000	1.000	$+\infty$	1.000
Стеганограма	Медіанний фільтр	169.248	0.975	25.846	0.978
	Фільтр Гауса	37.287	0.995	32.417	0.994
	Вейвлет стиснення	**26.105**	**0.998**	**34.014**	**0.997**
Стеганодані	Медіанний фільтр	148.967	0.966	26.406	0.967
	Фільтр Гауса	177.631	0.962	25.683	0.962
	Вейвлет стиснення	**1281.3**	**0.647**	**7.055**	**0.550**

Суттєві зміни значень МЯ ЦЗ для стеганограми, сформованої згідно методу *UNIWARD*, при використанні стандартних методів АСА (табл. 6.21), свідчать про появу візуальних спотворень ЦЗК, що демаскує факт проведення атаки на стеганосистему зв'язку. Застосування вейвлет-стиснення для обробки стеганограми дало можливість суттєво підвищити спотворення прихованих повідомлень при забезпеченні відносно малих змін значень МЯ для ЦЗК (рис. 6.23).

Отже, можемо зробити висновок, що використання ЗСД дозволяє спростити процедуру вибору оптимальних методів деструкції прихованих повідомлень — максимізувати ступінь спотворення стеганоданих при мінімальних візуальних змінах зображення-контейнера, — що представляє особливий інтерес при проведенні потокової обробки ЦЗ у системах моніторингу й контролю інфокомунікаційних систем.

6.8.2. Розробка комплексу прикладних програм для проведення пасивного стеганоаналізу цифрових зображень

Для автоматизації дослідження ефективності роботи структурних, статистичних та універсальних стеганодетекторів був розроблений комплекс прикладних програм, структурна схема якого наведена на рис. 6.24.

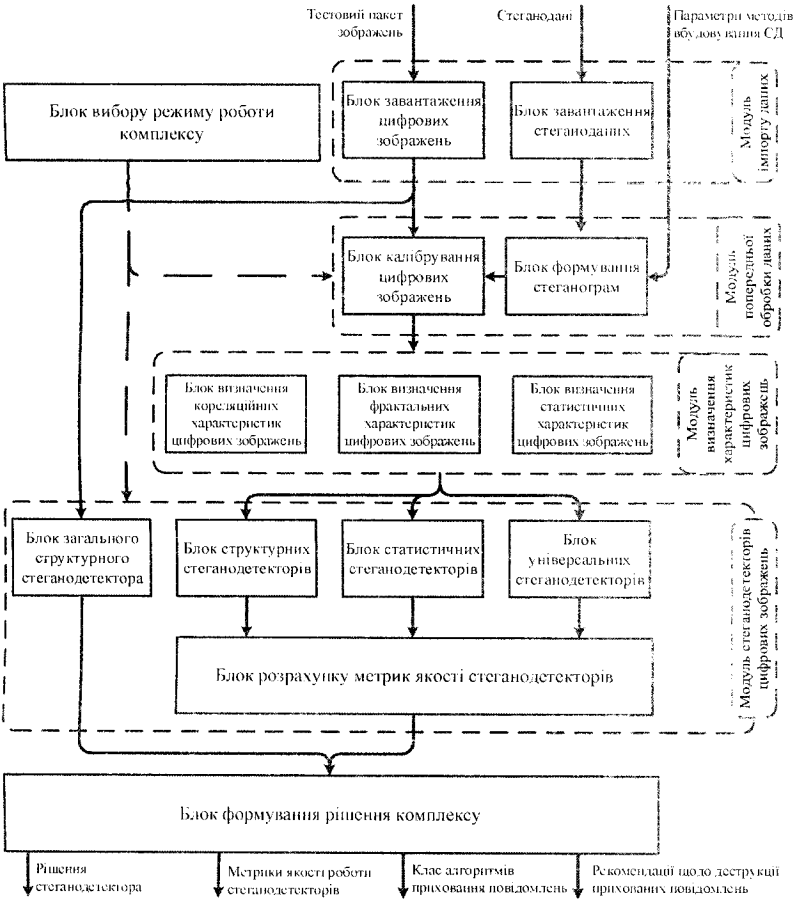


Рис. 6.24. Структурна схема комплексу прикладних програм для пасивного стеганоаналізу цифрових зображень

Обробка цифрових зображень з використанням розробленого програмного комплексу проводиться в декілька етапів (рис. 6.24). На першому етапі у блоці вибору режиму роботи комплексу обирається спосіб обробки ЦЗ:

1) режим налаштування стеганодетекторів — використовується для визначення параметрів останніх із застосуванням заданого тестового пакета ЦЗ і стеганограм, сформованих згідно одноетапних, багатоетапних і комплексних методів;

2) режим тестування стеганодетекторів — використовується для оцінки ймовірності виявлення стеганограм, сформованих згідно заданих або апріорно невідомих методів приховання повідомлень;

3) режим виявлення стеганограм — використовується при обробці інформаційних потоків в інфокомунікаційній системі для визначення наявності повідомлень, вбудованих до ЦЗК.

У залежності від вимог користувача, в якості стеганодетекторів можуть використовуватися структурні, статистичні або універсальні стеганодетектори, а також загальний структурний стеганодетектор.

На другому етапі задані ЦЗ $\boldsymbol{I}_{x,y}$ та представлені у вигляді кольорових зображень стеганодані $\boldsymbol{D}_{x,y}$ надходять до модуля імпорту даних, в якому здійснюється декомпресія заданих ЦЗК (блок завантаження цифрових) і СД (блок завантаження стеганоданих), представлених у графічних форматах стиснення зі втратами. Отримані матриці яскравості пікселів зображення-контейнера $\boldsymbol{I}_{x,y}$ та стеганоданих $\boldsymbol{D}_{x,y}$, а також параметри стеганографічних методів (ступінь заповнення ЦЗК стеганоданими Δ_C, ваговий параметр G) передають до блока формування стеганограм. Зображення-контейнери $\boldsymbol{I}_{x,y}$ та отримані на їхній основі стеганограми $\boldsymbol{S}_{x,y}$ передаються до блока калібрування ЦЗ, в якому згідно методу Фрідріх розраховуються калібровані зображення [172, 173] — відповідно, $\boldsymbol{I}^C_{x,y}$ та $\boldsymbol{S}^C_{x,y}$.

У випадку проведення ПСА ЦЗ з використанням структурних стеганодетекторів та загального структурного стеганодетектора калібрування $\boldsymbol{I}_{x,y}$ зображення-контейнера й стеганограми $\boldsymbol{S}_{x,y}$ не проводиться. При використанні ЗСД зображення-контейнери передаються до блока загального структурного стеганодетектора, в якому й проводиться їхня подальша обробка згідно алгоритму, наведеного на рис. 6.22.

На третьому етапі зображення-контейнери ($\boldsymbol{I}_{x,y}$, $\boldsymbol{I}^C_{x,y}$) та стеганограми ($\boldsymbol{S}_{x,y}$, $\boldsymbol{S}^C_{x,y}$) передаються до модуля визначення характеристик ЦЗ. До складу даного модуля входять блоки розрахунку кореляційних, фрактальних і статистичних характеристик цифрових зображень. Визначення кореляційних і фрактальних характеристик ЦЗ проводиться

з використанням представлених у [161] алгоритмів варіограмного, мультифрактального флуктуаційного та мультифрактального аналізів. Статистичні характеристики ЦЗ визначаються з використанням статистичних моделей зображень у просторовій (модель *SPAM*) і частотній (модель *CC-PEV*) областях, а також універсальній моделі *CDF*. Алгоритми обробки ЦЗ із застосуванням даних статистичних моделей наведені у [161].

На четвертому етапі розраховані кореляційні, фрактальні й статистичні характеристики зображень-контейнерів ($\boldsymbol{I}_{x,y}$, $\boldsymbol{I}^{c}_{x,y}$) і стеганограм ($\boldsymbol{S}_{x,y}$, $\boldsymbol{S}^{c}_{x,y}$) подаються до модуля стеганодетекторів, що складається зі структурних стеганодетекторів SD^{mod}_{VgA}, SD^{mod}_{MFDFA} і SD^{mod}_{MFA}, статистичних стеганодетекторів SD_{SPAM}, $SD_{CC\text{-}PEV}$ та SD_{CDF}, універсального стеганодетектора Авсібаша $SD_{Avcibaş}$ та загального структурного стеганодетектора. У залежності від обраного режиму роботи програмного комплексу, проводиться:

1) налаштування зазначених стеганодетекторів з використанням пакета тестових зображень і стеганограм згідно алгоритмів [161];

2) визначення ймовірності виявлення стеганограм, сформованих згідно заданих або апріорно невідомих стеганографічних методів.

Для інтегральної оцінки ефективності стеганодетекторів використані наступні метрики якості [184, 194, 195]:

1) загальна похибка класифікації (*Classification Accuracy*, *CA*) [184, 191]:

$$CA = \frac{TP + TN}{N_{ALL}}, \quad CA \in [0; 1],$$

де TP, TN — відповідно, кількість правильно класифікованих стеганограм та пустих ЦЗК; $N_{ALL} = (N_+ + N_-)$ — загальна кількість стеганограм (N_+) і незаповнених ЦЗК (N_-) у тестовому пакеті;

2) чутливість стеганодетектора (*Sensitivity*; *True Positive Rate*, *TPR*) — визначає кількість вірно класифікованих стеганодетекторами стеганограм відносно загальної кількості ЦЗ, що відносяться до класу стеганограм:

$$TPR = \frac{TP}{TP + FN}, \quad TPR \in [0; 1],$$

де FP, FN — відповідно, кількість помилок першого (хибне спрацювання) та другого (пропуск стеганограми) роду;

3) специфічність стеганодетектора (*Specifity*; *True Negative Rate*, *TNR*) — характеризує точність класифікації стеганодетектором пустих ЦЗ:

$$TNR = \frac{TN}{FP + TN}, \quad TNR \in [0; 1];$$

4) імовірність правильної класифікації стеганограм (*Precision*; *Positive Predictive Value*, *PPV*) — характеризує ймовірність правильного віднесенням стеганодетектором ЦЗ до класу стеганограм:

$$PPV = \frac{TP}{TP + FP}, \quad PPV \in [0; 1];$$

5) імовірність правильної класифікації незаповнених контейнерів (*Negative Predictive Value*, *NPV*) — використовується для оцінки ймовірності вірного віднесення стеганодетектором зображень до класу незаповнених ЦЗК:

$$PNV = \frac{TN}{TN + FN}, \quad PNV \in [0; 1];$$

6) площа під *ROC*-кривою (*Area under ROC*, *AUC*) — є інтегральною характеристикою точності класифікації стеганодетектором ЦЗК та розраховується згідно виразу (6.12);

7) ступінь впливу використовуваного простору ознак на точність детектування стеганограм (*Informedness*, *INs*) — розраховується згідно формули (6.13);

8) ступінь впливу використовуваного класифікатора на ефективність стеганодетектора (*Markedness*, *MNs*) — розраховується згідно виразу (6.14);

9) коефіцієнт кореляції Меттьюза (*Matthews Correlation Coefficient*, *MCC*) — використовується для оцінки величини кореляції рішень стеганодетектора та істинних міток класів досліджуваних ЦЗ та розраховується згідно формули (6.15).

На п'ятому етапі отримані метрики якості роботи структурних, статистичних та універсальних стеганодетекторів, а також результати роботи ЗСД (визначений клас методів формування стеганограм, рекомендації щодо деструкції прихованих повідомлень) передаються до блока обробки результатів ПСА цифрових зображень. У цьому блоці відбувається агрегація результатів роботи двокласових (бінарних) стеганодетекторів — наприклад, метрик якості їхньої роботи, — та загального структурного стеганодетектора, а також формується звіт щодо обробки заданого пакета тестових зображень.

6.9. Підсумки розділу

Переважна кількість сучасних моделей і методів ПСА ЦЗ грунтується на дослідженні та врахуванні лише спричинених приховуванням повідомлень змін статистичних характеристик власних шумів ЦЗК. Такий підхід призводить до істотного зниження ймовірності виявлення стеганограм у випадку вбудовування стеганографічних даних з використанням декількох компонентів графічних контейнерів. Для забезпечення високої вірогідності виявлення стеганограм незалежно від області прихування повідомлень нами запропоновано нову концепцію у цьому напрямку, засновану на використанні багаторівневої моделі зображення-контейнера й потужних методів структурного аналізу для виявлення слабких змін як ЦЗК загалом, так і його окремих компонентів, зумовлених вбудовуванням стеганоданих.

Враховуючи суттєве зниження ймовірності виявлення стеганограм з даними, вбудованими до ОПЦЗК, при використанні сучасних стеганодетекторів, удосконалено УСД Авсібаша. Для цього було запропоновано використати комплексне калібрування досліджуваних зображень на основі гаусівської, медіанної та вінерівської фільтрації, з підсумковим наслідком підсилення зміни характеристик ЦЗК, обумовлених використанням їхніх спеціальних перетворень для вбудовування СД, та підвищення точності виявлення стеганограм з даними, вбудованими до ОПЦЗК у випадку сильного заповнення стеганоданими.

На основі методів статистичного й стохастичного моделювання ЦЗ було розроблено багаторівневу модель зображень, що відрізняється від відомих моделей, використовуваних для проведення ПСА, врахуванням не тільки власних шумів, а й контурів і текстур об'єктів, присутніх на ЦЗК. Це дає можливість представити процес формування стеганограм, як внесення спотворень у декілька компонент графічних контейнерів, та, відповідно, використовувати відомі методи структурного аналізу цифрових сигналів для виявлення факту приховування повідомлень, визначення особливостей використаного стеганографічного методу (кількості етапів обробки ЦЗК, типу перетворень зображень контейнера та стеганоданих), а також оцінювання рівня заповненості ЦЗК стеганоданими й енергії прихованих повідомлень.

Виходячи з вимог щодо обчислювальної складності методів ПСА для застосування у системах моніторингу й контролю інфокомунікаційних систем реального часу, проведено модифікацію відомих методів структурного аналізу ЦЗ для створення швидких алгоритмів оцінювання їхніх параметрів. За рахунок раціонального поєднання порядкового й поколонного розподілів кореляційних і фрактальних характеристик зображень дістали подальшого розвитку варіограмний, флуктуаційний і мультифрактальний методи структурного аналізу ЦЗ, що дозволило

виявити слабкі зміни параметрів ЦЗК, обумовлені приховуванням повідомлень до ОПЦЗК.

Шляхом інтегрального застосування запропонованої багаторівневої моделі цифрових зображень і модифікованих методів структурного аналізу ЦЗ розроблено комплексний метод виявлення стеганограм з даними, вбудованими до ОПЦЗК, що відрізняється використанням як традиційних статистичних, так і кореляційних і фрактальних характеристик як зображень, так і їхніх компонентів. Це дозволило зменшити вимірність простору демаскуючих ознак стеганодетектора та, відповідно, скоротити тривалість обробки зображень при збереженні фіксованої точності виявлення стеганограм.

На основі запропонованих моделей і методів розроблено методику проведення ПСА ЦЗ та комплекс прикладних програм для підтримки її реалізації. Проведені експериментальні дослідження підтвердили високу точність детектування наявності прихованих повідомлень у ЦЗ (похибка віднесення ЦЗ до хибного класу не перевищує 5% навіть у випадку слабкого заповнення графічного контейнера стеганоданими) та ефективність застосування запропонованої методики для вибору цільових методів деструкції стеганограм, що дало можливість у 4 рази зменшити відношення СД/ЦЗК при мінімальних (не більше 2%) візуальних змінах підсумкових стеганограм.

ВИСНОВКИ

Як показує практика, актуальність проблеми інформаційної безпеки протягом попередніх років лише зростала, постійно стимулюючи при цьому пошук нових, більш дієвих методів захисту інформації. Збільшення попиту на ефективні системи захисту мультимедійного контенту, а також протидії прихованому обмінові даними, що можуть нести загрози здійснення несилового впливу на критичну інфраструктуру противника з метою отримання певного роду зисків у політичній, економічній або військовій сферах, є наслідком всім зрозумілої зацікавленості державних структур (яким, як правильно відзначено у [5], необхідно тепер протистояти не тільки розвідкам інших держав, але й внутрішнім супротивникам — адептам світового тероризму чи хакерства), а також зрозумілим бажанням керівників великих і малих фірм та, зрештою, пересічних громадян вберегти наявні в них конфіденційні дані від витоку, розголошення чи несанкціонованого використання. Вивчення існуючої наразі тенденції дає можливість стверджувати, що інтерес до впровадження й розвитку методів ефективного захисту мультимедійної інформації та аналізу безпечності інформаційного обміну лише нарощуватиме темпи свого зросту, чому, зокрема, сприятиме й бурхливий розвиток інформаційних технологій, який ми спостерігаємо протягом останніх десятиліть.

Зробити помітний внесок у загальну справу збереження державної таємниці, впевненості фірми у чесній грі конкурентів, а громадянина — у дієвості такого призабутого на сьогодні поняття як свобода особистості, поряд з іншими покликані й стеганографічні методи захисту й аналізу інформації, теоретичні й практичні основи якої викладено у пропонованій книзі. Одержані при цьому результати полягають в наступному:

- проведено аналіз спеціалізованих літературних джерел та ресурсів мережі *Internet* щодо перспективних напрямків, за якими можливе використання стеганографії як інструменту захисту інформації в автоматизованих системах обробки даних, що дозволило перейти до схеми стеганографічної системи, яка відповідає основним принципам теорії зв'язку;
- на підставі дослідження відомих публікацій вітчизняних і закордонних фахівців здійснено системне викладення проблем надійності й стійкості довільної стеганографічної системи по відношенню до видів здійснюваних на неї атак, а також оцінки пропускної здатності каналу прихованого обміну даними, яким, по суті, є стеганосистема. Виділено як спільні, так і характерні лише для стеганографічних систем види атак, на можливість існування яких необхідно зважати при організації чи

деструкції каналу прихованого зв'язку. Наведено результати існуючих інформаційно-теоретичних досліджень проблеми інформаційного приховання у випадку активної протидії порушника. У результаті було закладено обґрунтовану теоретичну базу для розробки комп'ютерних систем стеганографічного приховання конфіденційної інформації;

- викладені принципи, покладені до основи відомих на сьогодні стеганографічних методів, спрямованих на приховання конфіденційних даних у комп'ютерних файлах графічного, звукового й текстового форматів (розглянуто більше двадцяти методів);
- для зазначених методів проаналізовано особливості відповідних апаратів людини (зорового і слухового), зроблено акцент на характерні нюанси, що дозволяють скористатися існуючими обмеженостями ЗСЛ і ССЛ у стеганографічних цілях. Сформульовано загальні рекомендації стосовно можливих способів вбудовування бітів приховуваних повідомлень до контейнерів з метою підвищення рівня прихованості стеганоданих при застосуванні відомих алгоритмів стеганоаналізу;
- для демонстрації принципів, закладених у методи стеганографічного захисту інформації, наведено приклади комп'ютерних стеганосистем на їх основі, розроблених з використанням наочної математичної системи *Mathcad*. Усі етапи приховування супроводжені відповідними програмними модулями, загальна кількість яких перевищила 170. Одержані у ході моделювання результати проілюстровані значною кількістю графічного матеріалу. Проведено обчислення показників спотворення мультимедійних контейнерів при вбудовуванні до них бітів приховуваних даних, що дозволяє проводити аналіз оптимальності обрання того або іншого стеганометоду чи формату контейнера;
- розроблені системи дозволяють проводити стеганографічне приховання файлів будь-якого формату у файлах растрового зображення форматів *BMP*, *JPG*, *GIF*, в аудіофайлах форматів *WAV* та *MPEG*, а також у текстових файлах. Основні вимоги, що при цьому повинні виконуватися, — обрання файлу-контейнера належного об'єму і, зрештою, апаратна спроможність використовуваної обчислювальної системи.

Взаємна інтеграція стеганографічних і криптографічних технологій забезпечення інформаційної безпеки дає можливість позбутися слабких сторін, притаманних кожній з цих технологій поодинці, зробивши нові методи захисту інформації більш ефективними й оптимальними з аспектів обчислювальної складності та стійкості до зламу.

На завершення, сьогодні, коли з часу зародження комп'ютерної стеганографічної обробки мультимедійних даних пройшло близько чверті століття, ми можемо спостерігати за розквітом її використання у промислових масштабах.

Додаток А.
ОСНОВНІ ХАРАКТЕРИСТИКИ СИСТЕМИ АУДІОКОДУВАННЯ *MPEG/DAB*

А.1. Концепція стиснення звуку

Техніка кодування високоякісних аудіосигналів використовує усунення статистичної і психоакустичної надмірностей кодованого сигналу. Причому скорочення статистичної надмірності базується на врахуванні властивостей самих аудіосигналів, а психоакустичної – на врахуванні особливостей сприйняття звуку слуховою системою людини (ССЛ) [121,126–129,131–136].

Для усунення статистичної надмірності, зумовленої кореляційними зв'язками між сусідніми відліками часової функції звукового сигналу при його дискретизації, використовують алгоритми обробки без втрати інформації. Але при застосуванні останніх безпосередньо у часовому просторі швидкість цифрового потоку у порівнянні з її первинною величиною — приблизно 1.5 Мб/с:

$$44100 \text{ відл./с} \times 16 \text{ біт/відл.} \times 2 \text{ канали} = 1.4112 \text{ Мб/с},$$

— зменшується лише на 15...20% [135].

Психоакустична надмірність спричинена спектральними і часовими ефектами маскування [121,131,135,136]. Спектральне (частотне) маскування відбувається, коли нормально чутний за інших обставин звук перекривається іншим — більш гучним і близьким за частотою (рис. А.1). Гучний звук піднімає поріг чутливості ССЛ у деякому околі (пунктирна лінія), у результаті чого інший звук у цьому околі, який є тихим, але за звичайних обставин перевищує вказаний поріг, стає нечутним. Алгоритми аудіостиснення використовують цю властивість ССЛ і видаляють сигнали, що відповідають слабким (як початково нечутним, так і замаскованим) звукам, реалізуючи таким чином компресію із втратами. Часове маскування спостерігається, коли до або після гучного звуку певної частоти з достатньо малими інтервалом часу (зазвичай, не більше 5...10 мс) слідує більш тихий звук близької частоти (рис. А.2).

Скорочення надмірності представленням аудіосигналу у частотній області реалізується через кодування форми спектра, що дозволяє отримати найбільшу ефективність стиснення. Для цього використовують два способи: смугову фільтрацію (субсмугове представлення) й ортогональне перетворення (наприклад, Фур'є) [121,131,135].

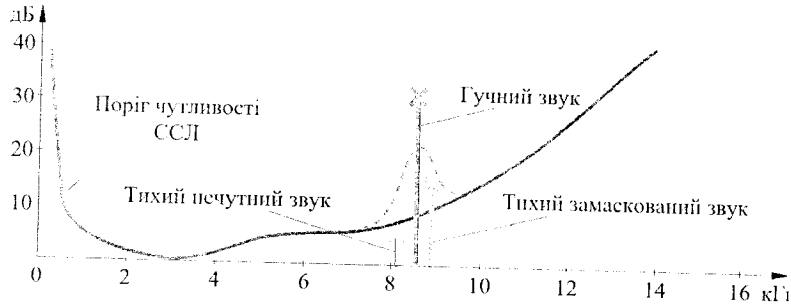


Рис. А.1. Поріг чутливості та спектральне маскування звуку

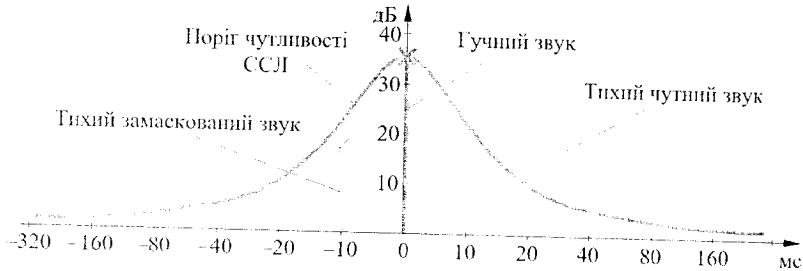


Рис. А.2. Поріг чутливості та часове маскування звуку

При субсмуговому представленні аудіосигнал пропускається через набір фільтрів, ширина смуг яких наближена до ширини критичних смуг ССЛ, і кодується окремо у кожній смузі. Як наслідок, статистична надмірність (надмірність опису) усувається за рахунок:

- відкидання частини субсмугових сигналів з нехтовно малими енергіями і зниження частоти дискретизації, використовуваної при кодуванні у кожній смузі (оскільки частота дискретизації визначається шириною смуги);
- різницевого представлення субсмугового сигналу, що досить повільно змінюється у часі;
- статистичного вирівнювання швидкості цифрового потоку — оскільки максимальний об'єм інформації, що відповідає передаванню повного набору субсмугових сигналів, зустрічається досить рідко, то існує можливість, використовуючи буферний пристрій, знизити середню швидкість передавання за рахунок певної (незначної) затримки сигналу.

Психоакустична надмірність сигналу усувається внаслідок:

- зниження точності представлення сигналу у кожній смузі — позаяк необхідне захисне відношення (інакше — відношення «сигнал/спо-

творення квантування») у межах критичної смуги не перевищує 30 дБ, то для опису сигналу достатньо 5 розрядів замість первинних 16-ти для ІКМ-сигналу;

- відкидання частини інформації, що не сприймається ССЛ через частотне і часове маскування у слуховому аналізаторі.

У разі представлення за допомогою ортогональних перетворень, аналіз спектру аудіосигналу, у порівнянні з субсмуговим кодуванням, здійснюється більш точно, наближуючись до представлення сигналу на рівні спектральних складових. Це робить використання при кодуванні закономірностей сприйняття останніх більш обґрунтованим. За такого представлення, сигнал фільтрується набором фільтрів, ширина смуги яких визначається відношенням частоти дискретизації до кількості точок на довжині вибірки і, з урахуванням використовуваної віконної функції, зазвичай складає 100...200 Гц при часі на аналіз 10...20 мс. При цьому статистична надмірність усувається шляхом:

- відкидання частини коефіцієнтів розкладу з малою енергією (оскільки будь-яке ортогональне перетворення концентрує енергію сигналу в обмеженому малому їх наборі);
- різницевого (диференціального) представлення коефіцієнтів розкладу;
- статистичного вирівнювання швидкості цифрового потоку.

Психоакустична надмірність сигналу при використанні ортогонального перетворення усувається в результаті:

- зниження точності представлення коефіцієнтів розкладу;
- відкидання частини коефіцієнтів, що не сприймаються через частотне і часове маскування у слуховому аналізаторі ССЛ.

А.2. Характеристики аудіосигналу

Основними характеристиками вхідного аудіосигналу є [137]:

- ширина смуги пропускання звуку: при частоті дискретизації $f_д = 48$ кГц вхідний аудіосигнал може займати весь діапазон НЧ до 20,3 кГц; при $f_д = 24$ кГц аудіосигнал, попередньо пропущений через ФНЧ перед понижуючою дискретизацією, займає обмежений діапазон — до 11.3 кГц;
- спряження по звуковій частоті: цифровий вхідний сигнал може відповідати параметрам спряження *AES/EBU* (*IEC 60958: Digital audio interface*);
- дискретність рівнів вхідного сигналу: система підтримує будь-яку дискретність до 22 біт/відлік включно;
- частота дискретизації вхідного аудіосигналу перед кодуванням у передавачі та цифрового вихідного аудіосигналу після декодування у приймачі складає 48 кГц. У випадку кодування за стандартом *MPEG*

Layer II з пониженою частотою дискретизації, на передавальному боці застосовується понижуюча дискретизація з 48 до 24 кГц, а на приймальній — підвищуюча — з 24 до 48 кГц. Це дає можливість уникнути необхідності одержання будь-якої відмінної від 48 кГц частоти взяття відліків при формуванні/використанні вхідного/вихідного ІКМ-аудіосигналу.

А.3. Характеристики аудіокодування

Визначальними характеристиками системи аудіокодування є [137]:

- аудіорежими:
  - одноканальний режим (для окремої монофонічної аудіопрограми);
  - стереорежим (лівий і правий канали для стереофонічної аудіопрограми);
  - двоканальний режим (для монофонічної аудіопрограми з двома ідентичними або ж двомовними каналами);
  - режим сполученого (інтенсивного) стерео (при цьому до певної частоти кодуються обидва канали, а вище — лише один);
- швидкість передавання:
  - згідно [126,127], дозволені швидкості передавання при $f_д = 48$ кГц в одноканальному режимі становлять: 32, 48, 56, 64, 80,96, 112, 128, 160 і 192 кбіт/с. Для решти 3-х режимів швидкість передавання подвоюється;
  - згідно [128,129], для будь-якого з 4-х режимів дозволені швидкості передавання кодованого аудіосигналу при $f_д = 24$ кГц становлять: 8, 16, 24, 32, 40, 48, 56, 64, 80, 96, 112, 128, 144 і 160 кбіт/с;
- довжина аудіокадру: окремий аудіокадр містить 1152 аудіовідліки ІКМ. Для частот дискретизації 48 і 24 кГц тривалість аудіокадру становить 24 і 48 мс відповідно. В останньому випадку аудіокадр є розділеним на дві частини однакової тривалості: парну (підкадр «0») і непарну (підкадр «1»), а поля пов'язаних з аудіопрограмою даних *(PAD)* оновлюються з інтервалом у 48 мс;
- максимальна розрядність аудіовибірки: 16 бітів.

А.4. Узагальнена структура аудіокодека MPEG-1(2)

Експертною групою *MPEG* (*Moving Pictures Expert Group*) надаються декілька профілів, що визначають область використання, та рівнів (*layers*), що обумовлюють ступінь стиснення цифрових даних. Розглянемо механізм кодування аудіосигналу *Musicam* (*MPEG Audio Layer II*, що використовується у стандартах *MPEG-1* і *2*).

Для частот дискретизації 48 і 24 кГц використовуються, відповідно, стандарти *ISO/IEC 11172-3* [126,127] та *ISO/IEC 13818-3* [128,129]. *Musicam* дозволяє зберігати аудіодані з імпульсно-кодовою модуляцією (ІКМ) у форматі 44,1 кГц/16 біт/стерео без відчутних втрат якості при швидкості цифрового потоку 192-256 кбіт/с, що становить 6...8-кратний виграш і відповідає рекомендаціям [130].

Слід зазначити, що [126–129] визначають лише кодований цифровий аудіопотік, а не побудову кодека, і тому будь-який кодек, спроможний згенерувати і коректно обробити стиснутий цифровий аудіопотік, може вважатися припустимим кодером *MPEG* [121,131]. Концептуальні засади аудіостиснення представлені також у [132–136].

Структура аудіокодера. Загальне уявлення про основні функції схеми (алгоритму) кодування звуку у відповідності до стандартів [126–129] можна отримати зі спрощеної структурної і блок-схеми аудіокодера, наведених, відповідно, на рис. А.3 і А.4.

Відліки первинного цифрового ІКМ-сигналу з частотою дискретизації $f_д$ подаються на вхід *MPEG*-кодера. Банк фільтрів аналізу (БФА) здійснює субсмуговий аналіз на основі дискретного косинусного перетворення (ДКП), розділяючи вхідний широкосмуговий сигнал на 32 послідовні рівновіддалені субсмугові складові (фільтровані відліки) з частотою дискретизації $f_д/32$ для кожної. Оскільки окремий аудіокадр утворює вибірка, яка складається з 1152 (для *Layer II*) відліків, то в кожній субсмузі знаходиться 1152/32=36 відліків або 12 гранул — згрупованих по 3 послідовних субсмугових відліки.

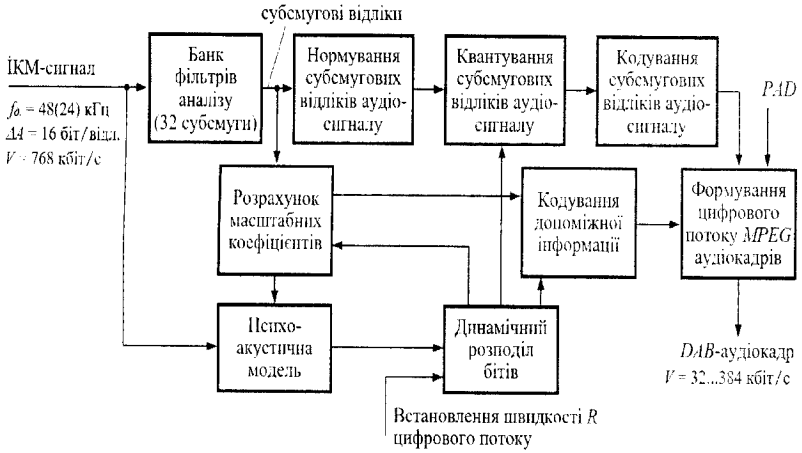


Рис. А.3. Структурна схема аудіокодера MPEG-1(2) [126–129]

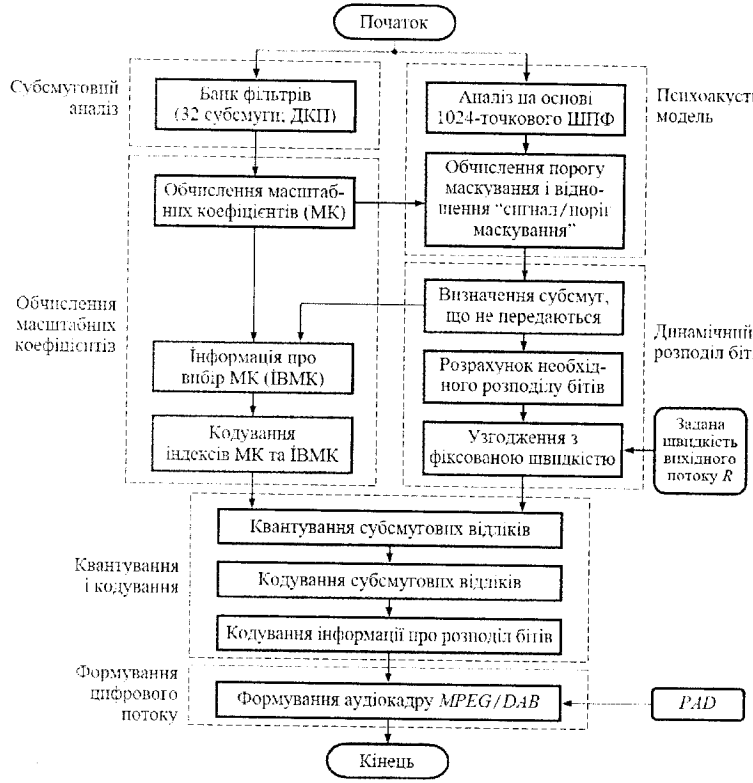


Рис. А.4. Блок-схема операцій аудіокодера *MPEG-1(2)* [126–129]

Перед квантуванням відліки, отримані з виходу БФА, масштабуються. При цьому 36 відліків аудіосигналу кожної з 32-х субсмуг розбиваються на 3 послідовних групи по 12 відліків, для кожної з яких здійснюється обчислення власного масштабного коефіцієнта (МК). Серед 12 відліків кожної групи визначається один з максимальним по модулю значенням, яке квантується шляхом порівняння з визначеним у [126–129,137] набором табличних значень МК, з яких обирається найближчий більший. Квантоване значення максимального відліку використовується в якості МК групи. Кожній субсмузі відповідає $36/12=3$ МК (по одному на групу), а загальна кількість МК аудіокадру — $3\cdot32=96$. При цьому кодується і передається не саме значення, а відповідний МК 6-бітовий індекс (ІМК), що може приймати значення від 0 до 62. Крім того, ІМК передаються лише для субсмуг, на які при розподілі було виділено біти для кодування відліків.

На початку кодування ІМК обчислюються дві різниці індексів сусідніх за групами МК: $P_1 = ІМК_1 - ІМК_2$ і $P_2 = ІМК_2 - ІМК_3$, яким, згідно їх значення, присвоюється клас від 1-го по 5-й. Пара значень класів різниць P_1 і P_2 однозначно вказує — скільки з трьох відповідних групі МК необхідно передати: один, два, три, максимальний з трьох або ж узагалі жодного (якщо відліки даної субсмуги не кодуються). Для цього визначається поточна конфігурація передавання МК і 2 бітовий код інформації про вибір масштабних коефіцієнтів (ІВМК), яка визначає номер і позицію МК у кожній субсмузі. Передається лише ІВМК субсмуг, на які виділено біти на кодування відліків. Розрахунок значень МК і обрання їх кількості для передавання декодеру виконується у блоці розрахунку МК (рис. А.3).

Паралельно з ДКП вхідний ІКМ-сигнал проходить обробку у блоці психоакустичної моделі (ПАМ), що на першому етапі реалізує швидке перетворення Фур'є (ШПФ). У роботі аудіокодера *MPEG* використовується два різних переведення з часової області у частотну: ДКП і ШПФ. Це зумовлене наближеним (грубим) розділенням по частоті вихідних продуктів ДКП, що для проведення психоакустичного аналізу, на відміну від продуктів прямого ШПФ, не підходить. Надалі у ПАМ для кожної субсмуги на виході БФА обчислюється ледь відчутний рівень шуму — мінімальний поріг маскування, — який використовується у процедурі розподілу бітів при визначенні для кожної з субсмуг відповідного квантувача. Кінцевим результатом ПАМ є визначення для кожної субсмуги відношення рівня сигналу до порогу маскування *SMR* (*Signal-to-Mask Ratio*), що дозволяє отримати шум квантування (різницю між первинним сигналом і його квантованим образом) невідчутним ССЛ — тобто нижчим за поріг маскування.

Базова модель адаптивного динамічного розподілу бітів (ДРБ) передбачає на своєму початку визначення загальної кількості доступних бітів і обчислення для кожної субсмуги відношення порогу маскування до рівня шуму *MNR* (*Mask-to-Noise Ratio*) як різниці (при оперуванні децибелами потужності) між відношенням рівня сигналу до рівня шуму *SNR* (*Signal-to-Noise Ratio*) і відношенням рівня сигналу до порогу маскування *SMR* (*Signal-to-Mask Ratio*). Значення *SNR* зазначене у стандарті, а *SMR* обчислюється у ПАМ. Кількості бітів для кодування відліків, ІМК та ІВМК кожної з субсмуг попередньо встановлюються у нуль і запускається ітераційна процедура розподілу наявних бітів між передаваними субсмугами. На кожному етапі ітерації збільшується розрядність відліку тієї субсмуги, *MNR* якої мінімальний.

У *Layer II* використовується рівномірне квантування субсмугових відліків. Кількість кроків квантування для кожної субсмуги задається таблицею [126–129], зміст якої залежить від швидкості передавання і

частоти дискретизації, значення яких визначають також номери субсмуг, на кодування відліків яких біти не виділяються. Для кожної субсмуги дозволено лише обмежену кількість можливих квантувань.

Очевидно, що ДРБ на основі ПАМ при стисненні вхідного ІКМ-сигналу й усунення притаманної останньому надмірності матиме наслідком психоакустичну диференціацію субсмугових відліків за кількістю кроків квантування і кінцеве зашумлення «чистого» вхідного сигналу, у наслідок чого кореляційний зв'язок між наймолодшими бітами кодових комбінацій відліків буде зруйнованим.

Для передавання інформації про субсмуговий ДРБ кодується відповідний використовуваному квантувачу індекс, що може приймати значення від 0 (коли відліки даної субсмуги не кодуються і не передаються) до 15.

Формування аудіокадру. Наприкінці *MPEG*-стиснення формувач кадру аудіокодера об'єднує дані про ДРБ, ІВМК, МК, коди квантованих субсмугових відліків, а також вміст заголовку і декілька слів *CRC* (*Cyclic Redundancy Check*) для виявлення помилок в єдиний цифровий потік *MPEG Audio Layer II*. Формований потік ділиться на аудіокадри, кожен з яких відповідає 1152-м відлікам ІКМ-аудіосигналу (що еквівалентно тривалості 24 мс при $f_д$ = 48 кГц).

Кожен аудіокадр цифрового мовлення *DAB (Digital Audio Broadcasting)* у кінці містить певну кількість байтів для так званих *PAD* (*Program-Associated Data*) — синхронних до аудіосигналу даних (регулювання динамічного діапазону, міжнародний стандартний код запису, індикація музики чи мови, універсальний/європейський товарний код *UPC/EAN*, динамічний текстовий заголовок, додатки користувача). Байти *PAD*, розмішені у послідовних аудіокадрах, формують канал *PAD*.

Два останніх байти аудіокадру виділено під фіксовані (*fixed*) дані *F-PAD* з дуже низькою швидкістю передавання. Канал *PAD* може також містити поле розширених (*extended*) даних *X-PAD* довільної, кратної 1 байту, довжини (зазвичай, до 16 байт), яка, проте, вимагає відповідного зменшення довжини поля аудіоданих, що матиме своїм наслідком погіршення якості звучання основного (звукового) контенту.

Структура аудіодекодера. Типові структурна і блок-схема аудіодекодера *MPEG/DAB* представлені на рис. А.5 і А.6 відповідно. Після синхронізації та читання заголовку аудіокадру визначаються необхідні для декодування параметри: аудіорежим, частота дискретизації, рівень організації контролю помилок тощо. Також здійснюється аналіз поля службових даних на наявність полів *F(X)-PAD*. На підставі отриманих відомостей цифровий потік розщеплюється на аудіопотік і потік *PAD*. Останній, після відповідної обробки, виводиться на графічний або знакосинтезуючий відтворювач інформації.

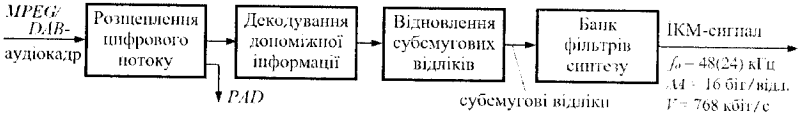


Рис. А.5. Структурна схема аудіодекодера *MPEG-1(2)* [126–129]

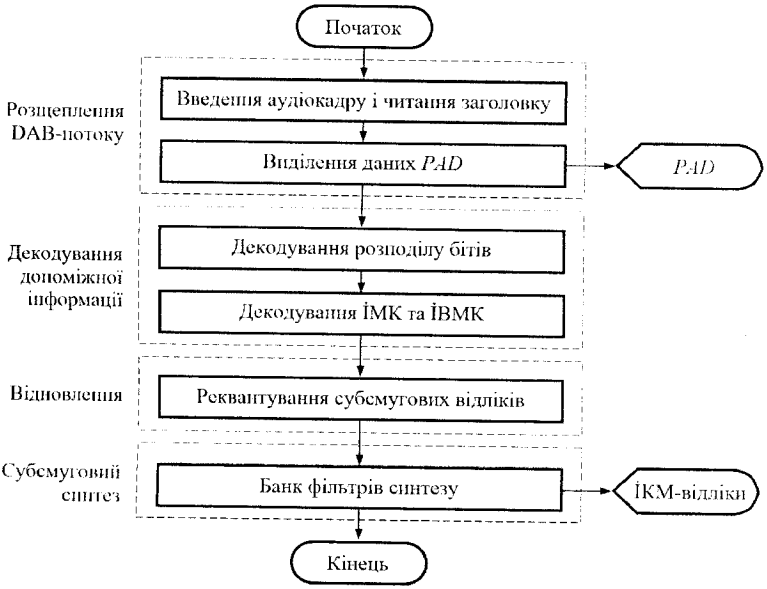


Рис. А.6. Блок-схема операцій аудіодекодера *MPEG-1(2)*

Для аудіопотоку проводиться декодування ДРБ — згідно стандартних таблиць визначається кількість бітів, виділена на кожен субсмуговий відлік, кількість рівнів квантування для субсмуг, коефіцієнти квантування, наявність чи відсутність групового кодування гранул, номер верхньої субсмуги, починаючи з якої і вище ДРБ не відбувався. Далі декодується ІВМК та ІМК. Згідно з ІВМК для кожної субсмуги визначається конфігурація передаваних МК, а для кожного 12-відлікового блоку субсмуги — свій ІМК, за яким з стандартної таблиці обирається *необхідне значення МК*.

З аудіокадру зчитуються 3 послідовні субсмугових відліки (гранула), і відповідно до того, з якою кількістю кроків вони були квантовані та чи було при кодуванні використане групування у гранули, здійснюється їх реквантування. Усі 1152 відновлені субсмугові відліки (включаючи непередані) пропускаються через банк вихідних фільтрів синтезу (БФС),

зворотному по функціях вхідному БФА, в яких 36 одночасних відліків кожної з 32-х субсмуг переводяться назад у 36 послідовних широкосмугових відліків — таким чином одержується 1152 послідовних ІКМ-аудіовідліки, придатні для подальшого цифро-аналогового перетворення (ЦАП).

Принципова відмінність між структурою ІКМ і стиснутих за алгоритмом *MPEG Layer II* аудіосигналів полягає у кодовому представленні відліків-результатів не у часовій (як для ІКМ), а в субсмуговій частотній, психоакустично обумовленій області [127,129], що позначається на форматі вихідних аудіокадрів й унеможливлює безпосереднє застосування методів стеганографічного приховання даних у цифрових аудіопотоках, орієнтованих на використання в якості контейнерів саме ІКМ-відліків.

А.5. Деякі стандартизовані таблиці кодування MPEG

Таблиця А.1

Масштабні коефіцієнти нормування та їх індекси

ІМК	Масштабний коефіцієнт	ІМК	Масштабний коефіцієнт	ІМК	Масштабний коефіцієнт
0	2.00000000000000	21	0.01562500000000	42	0.00012207031250
1	1.58740105196820	22	0.01240157071850	43	0.00009688727124
2	1.25992104989487	23	0.00984313320230	44	0.00007689947814
3	1.00000000000000	24	0.00781250000000	45	0.00006103515625
4	0.79370052598410	25	0.00620078535925	46	0.00004844363562
5	0.62996052494744	26	0.00492156660115	47	0.00003844973907
6	0.50000000000000	27	0.00390625000000	48	0.00003051757813
7	0.39685026299205	28	0.00310039267963	49	0.00002422181781
8	0.31498026247372	29	0.00246078330058	50	0.00001922486954
9	0.25000000000000	30	0.00195312500000	51	0.00001525878906
10	0.19842513149602	31	0.00155019633981	52	0.00001211090890
11	0.15749013123686	32	0.00123039165029	53	0.00000961243477
12	0.12500000000000	33	0.00097656250000	54	0.00000762939453
13	0.09921256574801	34	0.00077509816991	55	0.00000605545445
14	0.07874506561843	35	0.00061519582514	56	0.00000480621738
15	0.06250000000000	36	0.00048828125000	57	0.00000381469727
16	0.04960628287401	37	0.00038754908495	58	0.00000302772723
17	0.03937253280921	38	0.00030759791257	59	0.00000240310869
18	0.03125000000000	39	0.00024414062500	60	0.00000190734863
19	0.02480314143700	40	0.00019377454248	61	0.00000151386361
20	0.01968626640461	41	0.00015379895629	62	0.00000120155435

Таблиця А.2

Класи різниць індексів масштабних коефіцієнтів

Клас різниці	Різниця $P_i = ІМК_i - ІМК_{i+1}$ при $i = \{1,2\}$
1	$P_i \le -3$
2	$-3 < P_i < 0$
3	$P_i = 0$
4	$0 < P_i < 3$
5	$P_i \ge 3$

Таблиця А.3

Конфігурації передавання масштабних коефіцієнтів

Клас P_1	Клас P_2	Масштабні коефіцієнти, використовувані у кодері	Конфігурація передавання	Інформація про вибір масштабних коефіцієнтів (ІВМК)	Код ІВМК
1	1	1 2 3	1 2 3	0	«00»
1	2	1 **2 2**	1 2	3	«11»
1	3	1 **2 2**	1 2	3	«11»
1	4	1 **3 3**	1 3	3	«11»
1	5	1 2 3	1 2 3	0	«00»
2	1	**1 1** 3	1 3	1	«01»
2	2	**1 1 1**	1	2	«10»
2	3	**1 1 1**	1	2	«10»
2	4	**4 4 4**	4	2	«10»
2	5	**1 1** 3	1 3	1	«01»
3	1	**1 1 1**	1	2	«10»
3	2	**1 1 1**	1	2	«10»
3	3	**1 1 1**	1	2	«10»
3	4	**3 3 3**	3	2	«10»
3	5	**1 1** 3	1 3	1	«01»
4	1	**2 2 2**	2	2	«10»
4	2	**2 2 2**	2	2	«10»
4	3	**2 2 2**	2	2	«10»
4	4	**3 3 3**	3	2	«10»
4	5	1 2 3	1 2 3	0	«00»
5	1	1 2 3	1 2 3	0	«00»
5	2	1 **2 2**	1 2	3	«11»
5	3	1 **2 2**	1 2	3	«11»
5	4	1 **3 3**	1 3	3	«11»
5	5	1 2 3	1 2 3	0	«00»

Передаються лише МК, що винесені до стовпця «Конфігурація передавання». Позначення «1», «2» або «3» відповідно означають, що 1-й, 2-й або 3-й МК передаються у складі аудіокадру. Позначення «4» означає, що передається максимальний з МК. Якщо два чи три МК у межах субсмуги є однаковими, передається лише один з них.

Таблиця А.4

Коефіцієнти квантування

Кількість кроків квантування, N_{step}	Коефіцієнт A	Коефіцієнт B
3	0.750000000	–0.250000000
5	0.625000000	–0.375000000
7	0.875000000	–0.125000000
9	0.562500000	–0.437500000
15	0.937500000	–0.062500000
31	0.968750000	–0.031250000
63	0.984375000	–0.015625000
127	0.992187500	–0.007812500
255	0.996093750	–0.003906250
511	0.998046875	–0.001953125
1 023	0.999023438	–0.000976563
2 047	0.999511719	–0.000488281
4 095	0.999755859	–0.000244141
8 191	0.999877930	–0.000122070
16 383	0.999938965	–0.000061035
32 767	0.999969482	–0.000030518
65 535	0.999984741	–0.000015259

А.6. Базова модель процедури динамічного розподілу бітів

Для узгодження з фіксованою швидкістю передавання $\boldsymbol{R}$ попередньо визначається кількість вільних бітів $\boldsymbol{Б}^{\{a\}}_{вільн.}$, доступних для кодування відліків субсмуг аудіокадру $\boldsymbol{a}$, а також відповідних відлікам МК (максимум — трьох ІМК) та ІВМК. Це здійснюється відніманням від загальної кількості бітів, виділених на аудіокадр ($\boldsymbol{Б}^{\{a\}}_{\Sigma}$), бітів, необхідних для здійснення ДРБ відліків ($\boldsymbol{Б}^{\{a\}}_{розп.}$) і передавання службових або інших супутніх даних ($\boldsymbol{Б}^{\{a\}}_{с.о.}$):

$$\max(\boldsymbol{Б}^{\{a\}}_{вільн.}) = \boldsymbol{Б}^{\{a\}}_{а.о.} = \boldsymbol{Б}^{\{a\}}_{\Sigma} - \boldsymbol{Б}^{\{a\}}_{розп.} - \boldsymbol{Б}^{\{a\}}_{с.о.}. \qquad (А.1)$$

Принцип, що дотримується під час процедури ДРБ, полягає у мінімізації загального (для аудіокадру) відношення «рівень шуму/поріг маскування», але за обмеження, що загальна кількість бітів, використовуваних для кодування субсмугових відліків, ІМК, ІВМК та, зрештою, інформації про здійснення самого розподілу, не може перевищувати загальної кількості бітів, доступних для аудіоданих поточного аудіокадру ($\boldsymbol{Б}^{\{a\}}_{а.о.}$).

Процедура розподілу враховує як відліки на виході банку фільтрів, так і показники відношень рівнів сигналів до порогів маскування (*SMR – Signal-to-Mask Ratio*) на виході ПАМ. При цьому, для того щоб

одночасно задовольнити вимоги стосовно як бітової швидкості, так і порогу маскування, кількість бітів призначається для кожного відліку (або групи відліків) у кожній субсмузі окремо. У випадку низьких швидкостей, коли вимога, спричинена порогом маскування, не може бути дотримана, процедура розподілу намагається перерозподілити доступні біти по субсмугам, обґрунтованим ПАМ ССЛ.

У випадку $f_д$ = 48 кГц кількість кроків квантування, що використовується при дискретизації за рівнем відліків кожної з 32-х субсмуг, зведено до табл. А.5 і А.6.

Дані табл. А.5 застосовуються для стандартних бітових швидкостей від 56 до 192 кбіт/с в одноканальному та від 112 до 384 кбіт/с в інших аудіорежимах. При цьому біти не виділяються для кодування відліків субсмуг 27...31. Загальна кількість бітів, використовувана в окремому аудіокадрі для розподілу бітів відліків ($Б^{(a)}_{розп.}$), визначається підсумовуванням змісту стовпця « $Б^{(s)}_{розп.}$ » і у випадку табл. А.5 складає 88 бітів для одноканального режиму, 176 бітів для двоканального або стереорежиму і 132 біти для режиму комбінованого стерео.

Табл. А.6 використовується для бітових швидкостей 32 і 48 кбіт/с в одноканальному та 64 і 96 кбіт/с в інших аудіорежимах. При цьому біти не виділяються для кодування відліків субсмуг 8...31. Загальна кількість бітів, використовувана в окремому аудіокадрі для розподілу бітів відліків, становить 26 бітів для одноканального, 52 бітів для двоканального або стереорежиму і 40 бітів для режиму комбінованого стерео.

Три послідовних субсмугові відліки (так звана гранула) додатково можуть бути об'єднані (згруповані) для кодування. Кількість бітів, необхідна для формування кодових комбінацій (КК) про кількість кроків квантування окремих відліків, а також факт використання групового кодування представлено у табл. А.7. Якщо групування не потрібне, три відліки гранули кодуються окремими КК.

Розмір (а відповідно й кількість) кроків квантування відліків сигналу ($Δ_s$) для різних субсмуг може відрізнятися (див. табл. А.5 і А.6). Проте, для всіх 36-ти нормованих відліків сигналу всередині s-ї субсмуги величина $Δ_s$ залишається однаковою. Кількість кроків квантування при кодуванні субсмугових відліків складає 2^n-1 (де n — довжина кодового слова кількості кроків квантування відліку, див. табл. А.7) і може змінюватися дискретно від 3 (n = 2) до 65535 (n = 16). При цьому для економного витрачання бітів стандартом передбачено процедуру кодування гранул, відліки яких пройшли через квантувачі з 3, 5 або 9-ма кроками квантування, одним кодовим словом довжиною 5 (замість 6 при роздільному кодуванні), 7 (замість 9) і 10 (замість 12) бітів відповідно.

Таблиця А.5

Розподіл бітів і можливе квантування 36 відліків окремої субсмуги при $f_д$ = 48 кГц і високій бітовій швидкості ($R \geq 56$ кбіт/с на окремий канал)

Індекс квант.		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s	$Б_{гран}^{(s)}$	Кількість кроків квантування, $N_{квр}$															
0	4	–	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
1	4	–	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
2	4	–	3	7	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767	65535
3	4	–	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
4	4	–	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
5	4	–	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
6	4	–	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
7	4	–	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
8	4	–	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
9	4	–	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
10	4	–	3	5	7	9	15	31	63	127	255	511	1023	2047	4095	8191	65535
11	3	–	3	5	7	9	15	31	65535								
12	3	–	3	5	7	9	15	31	65535								
13	3	–	3	5	7	9	15	31	65535								
14	3	–	3	5	7	9	15	31	65535								
15	3	–	3	5	7	9	15	31	65535								
16	3	–	3	5	7	9	15	31	65535								
17	3	–	3	5	7	9	15	31	65535								
18	3	–	3	5	7	9	15	31	65535								
19	3	–	3	5	7	9	15	31	65535								
20	3	–	3	5	7	9	15	31	65535								
21	3	–	3	5	7	9	15	31	65535								
22	3	–	3	5	7	9	15	31	65535								
23	2	–	3	5	65535												
24	2	–	3	5	65535												
25	2	–	3	5	65535												
26	2	–	3	5	65535												
27	0	–															
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
31	0	–															

Таблиця А.6

Розподіл бітів і можливе квантування 36 відліків окремої субсмуги при $f_д$ = 48 кГц і низькій бітовій швидкості ($R \le 48$ кбіт/с на окремий канал)

Індекс квант.		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s	$Б_{розр.}^{(s)}$	Кількість кроків квантування, N_{step}															
0	4	–	3	5	9	15	31	63	127	255	511	1023	2047	4095	8191	16383	32767
1	4	–	3	5	9	15	31	63	127	255	511	1023	2047	4095	8191		
2	3	–	3	5	9	15	31	63	127								
3	3	–	3	5	9	15	31	63	127								
4	3	–	3	5	9	15	31	63	127								
5	3	–	3	5	9	15	31	63	127								
6	3	–	3	5	9	15	31	63	127								
7	3	–	3	5	9	15	31	63	127								
8	0	–															
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
31	0	–															

Таблиця А.7

Класи квантування субсмугових відліків

Кількість кроків квантування, N_{step}	Можливість групового кодування	Кількість відліків на одне кодове слово	Бітова довжина кодового слова, n
3	+	**3**	**5**
5	+	**3**	**7**
7	–	1	3
9	+	**3**	**10**
15	–	1	4
31	–	1	5
63	–	1	6
127	–	1	7
255	–	1	8
511	–	1	9
1 023	–	1	10
2 047	–	1	11
4 095	–	1	12
8 191	–	1	13
16 383	–	1	14
32 767	–	1	15
65 535	–	1	16

Таблиця А.8

Відношення сигнал/шум як функція кількості кроків квантування відліків

Кількість кроків квантування, N_{step}	*SNR*, дБ	Кількість кроків квантування, N_{step}	*SNR*, дБ	Кількість кроків квантування, N_{step}	*SNR*, дБ
0	0.00	31	31.59	2 047	67.98
3	7.00	63	37.75	4 095	74.01
5	11.00	127	43.84	8 191	80.03
7	16.00	255	49.89	16 383	86.05
9	20.84	511	55.93	32 767	92.01
15	25.28	1 023	61.96	65 535	98.01

Процедура розподілу бітів є ітераційною — на кожному циклі кількість кроків квантування відліків окремої субсмуги збільшується. У свою чергу, на кількість ітераційних циклів впливає ступінь важливості окремої субсмуги з точки зору ПАМ ССЛ.

На попередньому етапі для кожної субсмуги обчислюється відношення порогу маскування до рівня шуму (*MNR – Mask-to-Noise Ratio*):

$$MNR[\text{дБ}] = SNR[\text{дБ}] - SMR[\text{дБ}], \qquad (A.2)$$

де *SNR* (*Signal-to-Noise Ratio*) — відношення рівня сигналу до рівня шуму (знаходиться за табл. А.8); *SMR* (*Signal-to-Mask Ratio*) — відношення рівня сигналу до порогу маскування (обчислюється у ПАМ).

Далі необхідні кількості біт для кодування відліків ($\boldsymbol{Б}^{\{s\}}_{відл.}$), ІМК ($\boldsymbol{Б}^{\{s\}}_{імк}$) та ІВМК ($\boldsymbol{Б}^{\{s\}}_{івмк}$) кожної з субсмуг попередньо встановлюються у нуль, після чого запускається ітераційна процедура, кожен цикл якої складається з наступних кроків:

1) визначається найменше *MNR* серед усіх субсмуг ***a***-го кадру;

2) збільшується точність (кількість кроків) квантування відліків субсмуги, що має мінімальний *MNR* (наприклад, — ***s***-ї), шляхом переходу до квантування з наступним (більшим) індексом квантування (див. табл. А.5, А.6);

3) обчислюється нове значення *MNR* для даної (***s***-ї) субсмуги (формула (А.2));

4) оновлюється значення $\boldsymbol{Б}^{\{s\}}_{відл.}$ згідно додаткової кількості бітів, необхідних для квантування; якщо ненульова кількість бітів призначена ***s***-й субсмузі вперше, то згідно конфігурації передавання МК (див. табл. А.3) оновлюються значення кількостей біт для кодування ІВМК ($\boldsymbol{Б}^{\{s\}}_{івмк}$) та ІМК ($\boldsymbol{Б}^{\{s\}}_{імк}$);

5) обчислюється нове значення кількості вільних для кодування бітів:

$$\boldsymbol{Б}^{\{a\}}_{вільн.} = \boldsymbol{Б}^{\{a\}}_{а.о.} - \sum_{s}\left(\boldsymbol{Б}^{\{s\}}_{імк} + \boldsymbol{Б}^{\{s\}}_{імк} + \boldsymbol{Б}^{\{s\}}_{від.}\right). \qquad (А.3)$$

Процедура повторюється, поки значення $\boldsymbol{Б}^{\{a\}}_{вільн.}$ не стане меншим за будь-який з можливих приростів $\boldsymbol{Б}^{\{s\}}_{імк}$, $\boldsymbol{Б}^{\{s\}}_{імк}$ і $\boldsymbol{Б}^{\{s\}}_{від.}$ у межах наступного циклу.

А.7. Процедура квантування і кодування субсмугових відліків

Перед квантуванням усі 12 послідовних відліків кожної з трьох груп, на які у процесі масштабування розбивалася кожна з 32-х субсмуг, нормуються шляхом ділення їх значень на МК, який відповідає їх групі. Одержані величини $\boldsymbol{X\Sigma}$ квантуються згідно наступної процедури:

1) обчислюється значення $\boldsymbol{\Xi} = \boldsymbol{A}\cdot\boldsymbol{X\Sigma} + \boldsymbol{B}$ (коефіцієнти квантування $\boldsymbol{A}$ і $\boldsymbol{B}$ є стандартизованими — див. табл. А.4);
2) розглядаючи результат $\boldsymbol{\Xi}$ як дійсне число, що у випадку своєї від'ємності потребує представлення у комплементарному (додатковому) двійковому коді, обирається $\boldsymbol{n}$ найстарших бітів (де $\boldsymbol{n}$ — кількість розрядів, необхідних для представлення кодового слова кількості кроків квантування окремого відліку — див. табл. А.7);
3) інвертується найстарший значущий біт (*MSb — Most Significant bit*) серед $\boldsymbol{n}$ обраних (це необхідне для уникнення випадку отримання послідовності з усіх «1», яка стандартно використовується в заголовку аудіокадру для синхронізації).

Три послідовні субсмугові відліки, що утворюють окрему гранулу, можуть бути об'єднані для кодування спільною комбінацією. Кількість кроків, з якою при цьому мають бути квантовані відліки, і чи взагалі використовується об'єднане кодування відліків гранули, визначає табл. А.7. У випадку, якщо групування елементів гранули необхідне, три послідовні субсмугові відліки кодуються однією спільною КК, що являє собою двійкове представлення цілого беззнакового числа v, яке пов'язане з квантованими відліками x, y, z через задані стандартом співвідношення:

$$v_3 = 9\cdot z + 3\cdot y + x \qquad (v_3 = 0...26);$$
$$v_5 = 25\cdot z + 5\cdot y + x \qquad (v_5 = 0...124);$$
$$v_9 = 81\cdot z + 9\cdot y + x \qquad (v_9 = 0...728).$$

А.8. Таблиці індексів заголовку аудіокадру

Таблиця А.9

Номінальні сумарні бітові швидкості на окрему аудіопрограму при $f_д$ = 48 кГц

ІШП	Бітова швидкість, кбіт/с	Дозволені аудіорежими	ІШП	Бітова швидкість, кбіт/с	Дозволені аудіорежими
«0000»	у *DAB* не використовується		«1000»	128	будь-які
«0001»	32	одноканальний	«1001»	160	будь-які
«0010»	48	одноканальний	«1010»	192	будь-які
«0011»	56	одноканальний	«1011»	224	усі, окрім одноканального
«0100»	64	будь-які	«1100»	256	усі, окрім одноканального
«0101»	80	одноканальний	«1101»	320	усі, окрім одноканального
«0110»	96	будь-які	«1110»	384	усі, окрім одноканального
«0111»	112	будь-які	«1111»	заборонений індекс	

Таблиця А.10

Номінальні сумарні бітові швидкості на окрему аудіопрограму при $f_д$ = 24 кГц

ІШП	Бітова швидкість, кбіт/с	ІШП	Бітова швидкість, кбіт/с
«0000»	у *DAB* не використовується	«1000»	64
«0001»	8	«1001»	80
«0010»	16	«1010»	96
«0011»	24	«1011»	112
«0100»	32	«1100»	128
«0101»	40	«1101»	144
«0110»	48	«1110»	160
«0111»	56	«1111»	заборонений індекс

Таблиця А.11

Номінальні частоти дискретизації для вхідного/вихідного ІКМ-аудіосигналу

Індекс частоти дискретизації	Частота дискретизації при *ISO/IEC 13818-3*	Частота дискретизації при *ISO/IEC 11172-3*
«00»	у *DAB* не використовується	
«01»	24 кГц	48 кГц
«10»	у *DAB* не використовується	
«11»	зарезервовано для майбутнього використання	

Таблиця А.12

Аудіорежими, що можуть бути обрані в аудіокодері

Індекс режиму	Аудіорежим	Індекс режиму	Аудіорежим
«00»	стерео	«10»	двоканальний
«01»	сполучене стерео	«11»	одноканальний

Таблиця А.13

Передавані індекси масштабних коефіцієнтів у залежності від ІВМК

Код ІВМК	Дія
«00»	Передаються усі три ІМК для частин сигналу (груп відліків субсмуги) 0, 1 і 2
«01»	Передаються два ІМК: перший відповідає частинам 0 і 1, другий — частині 2
«10»	Передається один ІМК, дійсний для всіх трьох частин
«11»	Передаються два ІМК: перший відповідає частині 0, другий — частинам 1 і 2

А.9. Контентна та бітова структура аудіокадру

А.9.1. Заголовок аудіокадру *DAB* (перші 4 байти кадру; рис. А.7) містить інформацію, необхідну для належної роботи декодера, і є ідентичним заголовку аудіокадру *MPEG* [126–129], утримуючи у собі наступні елементи:

- 12-бітову синхрогрупу «1111 1111 1111»;
- 1-бітовий ідентифікатор алгоритму аудіокодування («0» — за стандартом *ISO/IEC 13818-3*; «1» — *ISO/IEC 11172-3*);
- 2-бітовий індикатор потоку (завжди «10» — *Layer II*);
- 1 біт мітки захисту (завжди «0» — до цифрового аудіопотоку для сприяння виявленню помилок вводиться контрольна сума *CRC*);
- 4-бітовий індекс швидкості передавання (ІШП), що показує сумарну бітову швидкість незалежно від поточного аудіорежиму. ІШП визначається усіма бітами аудіокадру, зокрема: бітами заголовку, аудіовідліків, *PAD*, інформації виявлення помилок (див. табл. А.9 і А.10);
- 2-бітовий індекс частоти дискретизації (завжди «01» — у сукупності з ідентифікатором алгоритму кодування вказує на 24(48) кГц; див. табл. А.11);
- 1 біт заповнення (завжди «0» — заповнення при $f_д$ = 24(48) кГц не потрібне);
- 1 біт права особистого користування (не використовується і не розпізнається);
- 2-бітовий індекс аудіорежиму (див. табл. А.12);

- 2-бітовий індекс розширення аудіорежиму — вказує субсмуги, що формують сполучене (інтенсивне) стерео: «00» – 4...31, «01» – 8...31, «10» – 12...31, «11» – 16...31; решта перших субсмуг кодується у режимі звичайного стерео;
- 1 біт авторських прав на цифровий аудіопотік (якщо «0» — права не заявлені);
- 1-бітову мітку оригінальності («1») чи копії («0») цифрового потоку;
- 2-бітовий індекс передспотворень виразності (завжди «00» — не вносяться).

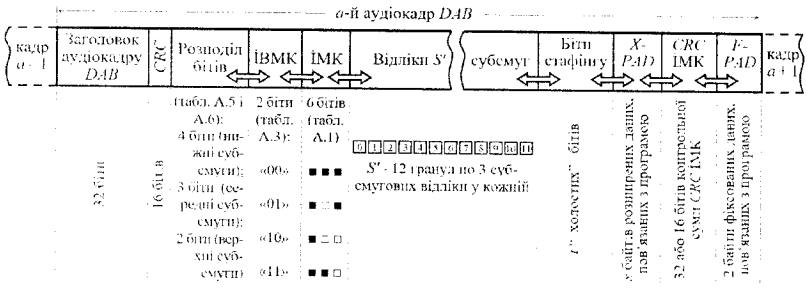


Рис. А.7. Бітова структура аудіокадру *DAB*

А.9.2. Для найбільш важливої і чутливої до помилок частини аудіоінформації всередині кодованого цифрового потоку (3 і 4-й байти заголовку кадру, дані про розподіл бітів, ІВМК) передбачається можливість контролю наявності помилок за допомогою 16-бітового слова парності *CRC*.

А.9.3. До складу аудіоданих входять (див. рис. А.7, А.8):

- поле розподілу бітів містить інформацію про квантувачі, використані при квантуванні відліків окремих субсмуг відповідних аудіоканалів, а також інформацію про потрійні послідовні відліки гранул, згруповані до спільних кодових слів, та інформацію про кількість бітів, використаних при кодуванні відліків. Зміст і довжина поля залежить від номера субсмуги, бітової швидкості і частоти дискретизації. Біти цього поля утворюють беззнакове ціле число, використовуване в якості індексу квантувача відповідної таблиці розподілу бітів (див., наприклад, табл. А.5 і А.6);
- поле 2-бітових ІВМК надає інформацію про кількість МК, що передаються для кожної субсмуги у відповідному аудіоканалі, а також — яким з трьох 12-відлікових частин сигналу (груп відліків) у поточному аудіокадрі вони відповідають (див. табл. А.13);

Рис. А.8. Контекстна структура аудіокадру цифрового мовлення *DAB* для стереорежиму («Л» і «П» — відповідно лівий і правий аудіоканали) [137]

- кратне 6 бітам поле ІМК інформує про коефіцієнти нормування, що на них мають бути помножені для реквантування прийняті субсмугові відліки відповідної частини сигналу кожної субсмуги аудіоканалу (див. табл. А.1);
- кодові комбінації окремих субсмугових відліків або гранул (див. табл. А.7). При цьому операція групування, що визначає кодування трьох послідовних відліків гранули окремої субсмуги аудіоканалу однією КК, виконується лише у випадку, якщо використовувана таблиця розподілу бітів для поточної субсмуги й індексу квантувача визначає кількість кроків квантування, що дорівнює 3, 5 або 9.

А.9.4. У *DAB*-аудіокадрах, що структурно є аналогічними кадрам *MPEG Audio Layer II*, поле службових даних зайняте 2 байтами фіксованих даних, пов'язаних з програмою (*F-PAD – Fixed Program-Associated Data*), і 16(32)-бітовою контрольною сумою *CRC* індексів масштабних коефіцієнтів, призначеною для захисту ІМК, що відносяться до субсмуг 2-х або 4-х різних спектральних груп усередині кодованого цифрового потоку. Крім того, за бажанням, може бути додана довільна кількість байт (від 4-х) розширених даних, пов'язаних з програмою (*X-PAD – eXtended Program-Associated Data*).

А.9.5. Загальна кількість бітів, доступних для аудіоданих в окремому кадрі *DAB*, визначається як

$$\boldsymbol{Б}^{\{a\}}_{a.o.} = \boldsymbol{t}_к \cdot \boldsymbol{R} - \left(\boldsymbol{Б}^{\{a\}}_{заг.} + \boldsymbol{Б}^{\{a\}}_{CRC} + \boldsymbol{Б}^{\{a\}}_{X\text{-}PAD} + \boldsymbol{Б}^{\{a\}}_{\boldsymbol{CRC}\,ІМК} + \boldsymbol{Б}^{\{a\}}_{F\text{-}PAD} \right), \qquad (А.4)$$

де $\boldsymbol{a}$ — індекс аудіокадру; $\boldsymbol{t}_к$ — тривалість аудіокадру (24 мс при $\boldsymbol{f}_д = 48$ кГц і 48 мс при $\boldsymbol{f}_д = 24$ кГц); $\boldsymbol{R}$ — бітова швидкість передавання аудіоданих (див. табл. А.9 і А.10); $\boldsymbol{Б}^{\{a\}}_{заг.} = 32$ — біти заголовку аудіокадру; $\boldsymbol{Б}^{\{a\}}_{CRC} = 16$ — біти контролю наявності помилок; $\boldsymbol{Б}^{\{a\}}_{X\text{-}PAD}$ — біти *X-PAD*; $\boldsymbol{Б}^{\{a\}}_{CRC\,ІМК} = 16(32)$ — біти контрольної суми *CRC* ІМК; $\boldsymbol{Б}^{\{a\}}_{\text{F-PAD}} = 16$ — біти *F-PAD*. Кількість бітів, що фактично займаються аудіоданими, може бути меншою, ніж обчислена за (А.4), — у цьому випадку між полем аудіоданих і полем *PAD* розміщуються біти стафінгу (холостого заповнення), що не несуть корисної інформації.

А.10. Поля пов'язаних з аудіопрограмою даних

Розміщення полів *F-PAD* і *X-PAD* у межах аудіокадру зображене на рис. А.9: поле *F-PAD* (байти L–1 і L кадру) йде одразу після контрольної суми ІМК; поле *X-PAD* розміщене між полями аудіоданих і *CRC* ІМК.

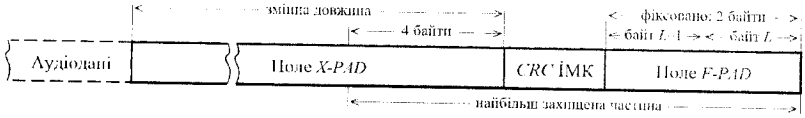


Рис. А.9. Розміщення байтів *PAD* у межах аудіокадру *DAB*

А.11. Реквантування і декодування субсмугових відліків

На першому етапі декодер синхронізується з бітовим аудіопотоком, що надходить. Одразу після запуску це може бути виконане шляхом пошуку у потоці кодової синхрогрупи з 12-ти «1». За певних застосувань (як, зокрема, й у ЦЗМ) такі елементи заголовку, як ідентифікатор стандарту аудіокодування, індикатор потоку, статус захисту від помилок, є заздалегідь відомими декодерові. Отже, в якості слова синхронізації можуть розглядатися перші 16 бітів заголовку, створюючи при цьому більш надійну синхронізацію.

А.11.1. Декодування інформації про динамічний розподіл. Різним комбінаціям швидкості передавання $\boldsymbol{R}$ і частоти дискретизації $f_д$ стандартами [126,127] поставлено у відповідність індивідуальні таблиці розподілу бітів (на зразок табл. А.5, А.6). Процедура декодування інформації з подібної таблиці складається з наступних трьох кроків:

1) читання інформації у 2, 3 або 4-х бітах розподілу $\boldsymbol{Б}^{[s]}_{розп.}$ (значення задане у другій колонці таблиці розподілу) для s-ї субсмуги, що інтерпретуються як код беззнакового цілого десяткового числа — індексу квантування q;
2) індекс квантування q разом з номером субсмуги s використовуються в якості координат входу до таблиці розподілу: значення, отримане на перетині колонки індексу квантування і рядка номеру субсмуги, представляє собою кількість кроків $N_{steps,q}$ квантування відліків у даній субсмузі;
3) використовуючи таблиці класів квантування (табл. А.7) і коефіцієнтів реквантування (табл. А.14) визначають бітову довжину n КК квантованого субсмугового відліку і коефіцієнти реквантування C, D. Крім того, виконують перевірку факту використання групового кодування.

А.11.2. Декодування ІВМК. 36 відліків у кожній субсмузі всередині аудіокадру розбиваються на 3 однакові групи (по 12 субсмугових відліків у кожній). Кожна група має свій власний МК. Кількість МК, що мають бути зчитані з бітового потоку, залежить від ІВМК

субсмуги s. 2-бітові коди ІВМК зчитуються з загального потоку лише для тих субсмуг, які характеризуються ненульовою кількістю бітів, виділених на їх кодування. Якщо ІВМК(s) = «00», то передаються три МК (по одному для кожної з 3-х груп відліків). Якщо ІВМК(s) = «01», то передаються два МК: 1-й відповідає групам 0 і 1, а 2-й — групі 2. Якщо ІВМК(s) = «10», передається лише один МК, що відповідає усім трьом групам відліків. Якщо ІВМК(s) = «11», передаються два МК: 1-й відповідає групі 0, а 2-й — групам 1 і 2.

Таблиця А.14

Коефіцієнти реквантування

Кількість кроків квантування, N_{step}	Коефіцієнт C	Коефіцієнт D
3	1.33333333333	0.50000000000
5	1.60000000000	0.50000000000
7	1.14285714286	0.25000000000
9	1.77777777777	0.50000000000
15	1.06666666666	0.12500000000
31	1.03225806452	0.06250000000
63	1.01587301587	0.03125000000
127	1.00787401575	0.01562500000
255	1.00392156863	0.00781250000
511	1.00195694716	0.00390625000
1 023	1.00097751711	0.00195312500
2 047	1.00048851979	0.00097656250
4 095	1.00024420024	0.00048828125
8 191	1.00012208522	0.00024414063
16 383	1.00006103888	0.00012207031
32 767	1.00003051851	0.00006103516
65 535	1.00001525902	0.00003051758

А.11.3. Декодування МК. Для кожної кодованої за результатом ДРБ субсмуги з бітового потоку зчитується код(-и) відповідного(-их) ІМК. Кількість кодованих МК і групи відліків, яким вони відповідають, визначаються через ІВМК(s). Кожні 6 бітів коду ІМК інтерпретуються як беззнакове ціле десяткове число з відповідної колонки табл. А.1. Після реквантування, на визначені за таблицею МК для регенерації рівня множиться кожен відлік значущої субсмуги.

А.11.4. Реквантування і регенерація субсмугових відліків. З загального бітового потоку зчитуються коди відліків, довжина n яких знаходиться за табл. А.7. У випадку, якщо КК відповідає гранулі з трьох послідовних відліків, додатково має бути виконане розгрупування. При цьому КК гранули розглядається як беззнакове ціле число v. Процедура одержання трьох окремих КК для відліків x, y, z (див. розділ А.6) полягає в наступному:

- десяткове число v ділиться на кількість кроків квантування $N_{кер}$ відліків гранули: залишок від ділення є десятковим кодом відліку x, а новим значенням v стає ціла частина від результату ділення;
- нове число v ділиться на $N_{кер}$: залишок від ділення є десятковим кодом відліку y, а v присвоюється ціла частина від результату ділення;
- число v ділиться на $N_{кер}$, залишок від ділення є десятковим кодом відліку z.

Перший біт кожної КК відліків (як тих, що були кодовані незалежно, так і тих, що кодувалися у гранулах і були згодом розгруповані) інвертується, а отриманий результат розглядається як дробове (дійсне) число, що представлене у додатковому двійковому коді. При цьому знаком числа є *MSb*.

Реквантовані значення відліків отримуються застосуванням лінійного виразу

$$\boldsymbol{X\Sigma = C\cdot(\Xi + D)}, \qquad \text{(A.5)}$$

де $\boldsymbol{X\Sigma}$ — реквантоване значення відліку; $\boldsymbol{\Xi}$ — дійсне десяткове число, отримане з двійкової КК відліку; $\boldsymbol{C}$ і $\boldsymbol{D}$ — задані стандартом коефіцієнти реквантування (див. табл. А.14).

Реквантовані значення регенеруються шляхом обчислення за виразом

$$\boldsymbol{V = X\Sigma\cdot MK}. \qquad \text{(A.6)}$$

А.12. Чутливість бітів аудіокадру до помилок

У [126,127] вказано чутливість окремих бітів зі структури аудіокадру до випадкових помилок (рівень впливу одиночного спотворення на загальну якість звучання) (табл. А.15). Кожному біту кадру ставиться у відповідність чутливість величиною від 0 до 5, відображаючи ступінь погіршення відновлення/звучання у результаті дії однієї локальної помилки (табл. А.16).

Таблиця А.15

Чутливість структурних бітів аудіопотоків *MPEG Layer II* до помилок

Компонент аудіокадру	Порядкові номери бітів	Ступінь чутливості
Заголовок	довільні	«5»
Розподіл бітів	довільні	«5»
ІВМК	довільні	«5»
ІМК	5*(MSb)*/4/3/2/1/0*(LSb)*	«4/4/4/3/2/1»
Субсмугові відліки	15*(MSb)*...8/7...5/4,3/2...0*(LSb)*	«3/2/1/0»

Таблиця А.16

Інтерпретація ступенів чутливості бітів аудіопотоків до помилок

Ступінь чутливості	Спотворення звучання у випадку помилки в окремому розряді з заданою чутливістю
«5»	катастрофічне (неможливість синтезу вихідного аудіопотоку)
«4»	надзвичайно подразнюючe слух
«3»	подразнюючe слух
«2»	несуттєво подразнюючe слух
«1»	чутне, але не подразнюючe слух
«0»	нечутне

Наведені значення виведені на підставі відомостей про структуру аудіокодека і відповідають випадку невикористання будь-яких схем виявлення помилок. Для захисту від пакетів помилок використовують часове перемежовування символів.

За відсутності завадостійкого кодування спотворення різних структурних одиниць аудіокадрів через дію зосереджених завад призводять до неоднакових рівнів погіршення якості звучання відтворюваного аудіопотоку. Тому найважливіші й чутливі до помилок компоненти аудіоінформації всередині кодованого цифрового потоку (заголовок кадру, дані ДРБ, ІВМК, ІМК) додатково захищають *CRC*-кодом. Захист поля субсмугових відліків в межах аудіокадру стандартами не передбачено.

Серед усіх складових поля аудіоданих найнижчу чутливість до помилок має бітовий контент субсмугових відліків: максимум — подразнюючe спотворення (3-й ступінь чутливості з 5 передбачених) у випадку впливу завади на один з найстарших бітів (при виділенні на кодування відліку 16 бітів). Спотворення трьох наймолодших бітів стандартом при цьому відноситься до нечутного (0-й ступінь чутливості).

Додаток Б.
ПСИХОАКУСТИЧНА МОДЕЛЬ № 1

Стандартизована ПАМ № 1 передбачає наступні кроки реалізації алгоритму [126,127]:

Б.1. Швидке перетворення Фур'є

Поріг маскування одержується з оцінки спектрального розподілу потужності, обчисленої шляхом 1024-точкового ШПФ ($N = 1024$) безпосередньо над ділянкою вхідного ІКМ-сигналу, виокремленої вікном Ганна (*Hann window*) $\boldsymbol{h}$ (рис. Б.1) — для зменшення спотворень, викликаних ефектом Гіббза (*Gibbs phenomenon*)):

$$h_i = \sqrt{8/3}\cdot\left[1-\cos\left(2\cdot\pi\cdot i/N\right)\right]/2, \quad i = 0\ldots N-1. \qquad \text{(Б.1)}$$

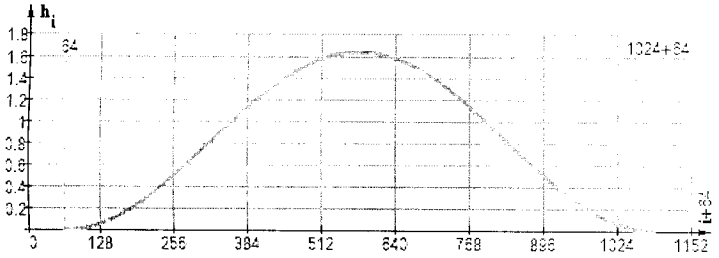


Рис. Б.1. Приклад вікна (вагової функції) Ганна

З аспектів синхронності (часового узгодження між розподілом бітів і відповідними їм субсмуговими відліками) ІКМ-відліки, що вводяться до ШПФ, мають бути затримані:

- затримка субсмугового БФА становить 256 відліків (або $256\cdot\Delta t = 5.333$ мс при частоті дискретизації $f_д = 48$ кГц), що відповідає зсуву вікна Ганна на 256 відліків;
- вікно Ганна має співпадати зі субсмуговими відліками аудіокадру. При $S_K = 1152$ відліки додатковий зсув вікна становить:

$$\Delta = (S_K - N)/2 = 64 \text{ відліки.}$$

Технічні характеристики ШПФ для *MPEG Layer II* є наступними:

- довжина перетворення: $N = 1024$;
- розмір вікна при $f_д = 48$ кГц: $N\cdot\Delta t = 21.333$ мс;

- розподільча здатність по частоті: $\Delta f = f_д / N = 46.875$ Гц/відл.;
- спектральний розподіл потужності (лінійчастий спектр):

$$X_k = 20 \cdot lg\left[\left|\frac{1}{N} \cdot \sum_{i=0}^{N-1} h_i \cdot x_i \cdot exp\left(i \cdot \frac{-j \cdot 2 \cdot \pi \cdot k}{N}\right)\right|\right] \text{ дБ}, \qquad (Б.2)$$

де x_i — i-й відлік вхідного ІКМ-сигналу; $k = 0...N/2$ — індекси коефіцієнтів ШПФ.

Для того щоб максимальний рівень звукового тиску відповідав 96 дБ, має бути здійснене додаткове нормування до опорного рівня у $20 \cdot lg(2^{16}) \approx 96{,}33$ дБ.

Б.2. Визначення рівня звукового тиску для кожної субсмуги

Рівень звукового тиску (енергія сигналу відліку) в s-й субсмузі кодування:

$$\begin{aligned} L_{cc_s} &= max\left[X_k,\ 20 \cdot lg\left(32768 \cdot MK_{max_s}\right) - 10\right] \approx \\ &\approx max\left[X_k,\ 20 \cdot lg\left(MK_{max_s}\right) + 80.309\right] \text{ дБ}, \end{aligned} \qquad (Б.3)$$

де X_k — рівень звукового тиску k-ї спектральної компоненти ШПФ, яка є максимальною за величиною у частотному діапазоні, що відповідає n-й субсмузі розкладу ШПФ; MK_{max_s} — максимальне значення серед 3-х МК s-ї субсмуги всередині аудіокадру; «–10 дБ» — елемент, що корегує різницю між піковим і середньоквадратичними (*RMS – Root-Mean-Square*) рівнями звукового тиску.

Б.3. Визначення порогу тиші (абсолютного порогу)

Значення порогу у тиші LTq, також відомого як абсолютний поріг (ATh – *Absolute Threshold*), а також критичних смужок (*CBR – Critical Band Rates*) стандартизовані [126,127] і визначаються частотою дискретизації вхідного ІКМ-сигналу — наприклад, для $f_д = 48$ кГц відповідає табл. Б.1. Значення доступні для кожного відліку у частотній області з обчисленим порогом маскування.

Для абсолютного порогу ATh здійснюється зсув ($offset$), що залежить від загальної швидкості передавання даних R результуючого стиснутого потоку: $offset = -12$ дБ для $R \geq 96$ кбіт/с і $offset = 0$ дБ для $R < 96$ кбіт/с на один канал:

$$LTq = ATh + offset. \qquad (Б.4)$$

Таблиця Б.1

Частоти, критичні смужки (***CBR***) та абсолютний поріг (***ATh***) для *MPEG Layer II* при частоті дискретизації $f_д$ = 48 кГц

Інд.	Частота, Гц	***CBR***, барк	***ATh***, дБ	Інд.	Частота, Гц	***CBR***, барк	***ATh***, дБ	Інд.	Частота, Гц	***CBR***, барк	***ATh***, дБ
1	46.88	0.463	42.10	*43*	2015.63	13.154	-0.32	*85*	6937.50	20.461	3.09
2	93.75	0.925	24.17	*44*	2062.50	13.302	-0.54	*86*	7125.00	20.616	3.33
3	140.63	1.385	17.47	*45*	2109.38	13.446	-0.75	*87*	7312.50	20.766	3.60
4	187.50	1.842	13.87	*46*	2156.25	13.586	-0.97	*88*	7500.00	20.912	3.89
5	234.38	2.295	11.60	*47*	2203.13	13.723	-1.20	*89*	7687.50	21.052	4.20
6	281.25	2.742	10.01	*48*	2250.00	13.855	-1.43	*90*	7875.00	21.188	4.54
7	328.13	3.184	8.84	*49*	2343.75	14.111	-1.88	*91*	8062.50	21.318	4.91
8	375.00	3.618	7.94	*50*	2437.50	14.354	-2.34	*92*	8250.00	21.445	5.31
9	421.88	4.045	7.22	*51*	2531.25	14.585	-2.79	*93*	8437.50	21.567	5.73
10	468.75	4.463	6.62	*52*	2625.00	14.807	-3.22	*94*	8625.00	21.684	6.18
11	515.63	4.872	6.12	*53*	2718.75	15.018	-3.62	*95*	8812.50	21.797	6.67
12	562.50	5.272	5.70	*54*	2812.50	15.221	-3.98	*96*	9000.00	21.906	7.19
13	609.38	5.661	5.33	*55*	2906.25	15.415	-4.30	*97*	9375.00	22.113	8.33
14	656.25	6.041	5.00	*56*	3000.00	15.602	-4.57	*98*	9750.00	22.304	9.63
15	703.13	6.411	4.71	*57*	3093.75	15.783	-4.77	*99*	10125.00	22.482	11.08
16	750.00	6.770	4.45	*58*	3187.50	15.956	-4.91	*100*	10500.00	22.646	12.71
17	796.88	7.119	4.21	*59*	3281.25	16.124	-4.98	*101*	10875.00	22.799	14.53
18	843.75	7.457	4.00	*60*	3375.00	16.287	-4.97	*102*	11250.00	22.941	16.54
19	890.63	7.785	3.79	*61*	3468.75	16.445	-4.90	*103*	11625.00	23.072	18.77
20	937.50	8.103	3.61	*62*	3562.50	16.598	-4.76	*104*	12000.00	23.195	21.23
21	984.38	8.410	3.43	*63*	3656.25	16.746	-4.55	*105*	12375.00	23.309	23.94
22	1031.25	8.708	3.26	*64*	3750.00	16.891	-4.29	*106*	12750.00	23.415	26.90
23	1078.13	8.996	3.09	*65*	3843.75	17.032	-3.99	*107*	13125.00	23.515	30.14
24	1125.00	9.275	2.93	*66*	3937.50	17.169	-3.64	*108*	13500.00	23.607	33.67
25	1171.88	9.544	2.78	*67*	4031.25	17.303	-3.26	*109*	13875.00	23.694	37.51
26	1218.75	9.805	2.63	*68*	4125.00	17.434	-2.86	*110*	14250.00	23.775	41.67
27	1265.63	10.057	2.47	*69*	4218.75	17.563	-2.45	*111*	14625.00	23.852	46.17
28	1312.50	10.301	2.32	*70*	4312.50	17.688	-2.04	*112*	15000.00	23.923	51.04
29	1359.38	10.537	2.17	*71*	4406.25	17.811	-1.63	*113*	15375.00	23.991	56.29
30	1406.25	10.765	2.02	*72*	4500.00	17.932	-1.24	*114*	15750.00	24.054	61.94
31	1453.13	10.986	1.86	*73*	4687.50	18.166	-0.51	*115*	16125.00	24.114	68.00
32	1500.00	11.199	1.71	*74*	4875.00	18.392	0.12	*116*	16500.00	24.171	68.00
33	1546.88	11.406	1.55	*75*	5062.50	18.611	0.64	*117*	16875.00	24.224	68.00
34	1593.75	11.606	1.38	*76*	5250.00	18.823	1.06	*118*	17250.00	24.275	68.00
35	1640.63	11.800	1.21	*77*	5437.50	19.028	1.39	*119*	17625.00	24.322	68.00
36	1687.50	11.988	1.04	*78*	5625.00	19.226	1.66	*120*	18000.00	24.368	68.00
37	1734.38	12.170	0.86	*79*	5812.50	19.419	1.88	*121*	18375.00	24.411	68.00
38	1781.25	12.347	0.67	*80*	6000.00	19.606	2.08	*122*	18750.00	24.452	68.00
39	1828.13	12.518	0.49	*81*	6187.50	19.788	2.27	*123*	19125.00	24.491	68.00
40	1875.00	12.684	0.29	*82*	6375.00	19.964	2.46	*124*	19500.00	24.528	68.00
41	1921.88	12.845	0.09	*83*	6562.50	20.135	2.65	*125*	19875.00	24.564	68.00
42	1968.75	13.002	-0.11	*84*	6750.00	20.300	2.86	*126*	20250.00	24.597	68.00

Наведена у табл. Б.1 залежність абсолютного порогу від частоти також представлена на рис. Б.2.

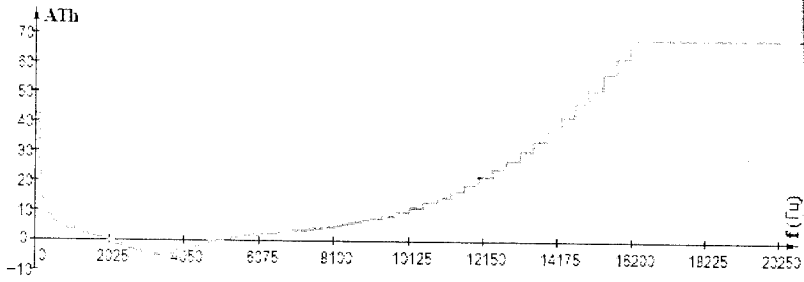


Рис. Б.2.

Б.4. Пошук тональних і нетональних компонент аудіосигналу

Тональність маскуючої компоненти (див. розділ А.1 Додатку) впливає на поріг маскування. Тому доцільно провести розрізнення між тональними (більш подібними до синусоїди) і нетональними (більш подібними до шуму) компонентами, видобування яких зі спектру ІКМ-сигналу на основі ШПФ дає можливість у подальшому обчислити глобальний поріг маскування.

Пошук вищезазначених компонент починається з визначення локального максимуму, після чого видобуваються тональні компоненти й обчислюється інтенсивність нетональних компонент усередині частотного діапазону критичних смужок. Межі критичних смужок (*CBB – Critical Band Boundaries*) задані стандартом [126,127] і для випадку, коли $f_д$ = 48 кГц представлені у табл. Б.2.

Таблиця Б.2

Межі критичних смужок (*CBB*) для *MPEG Layer II* при $f_д$ = 48 кГц
(значення частот відповідають верхнім значенням критичних смужок)

№	Індекс з табл. Б.1	Частота, Гц	Висота, барк	№	Індекс з табл. Б.1	Частота, Гц	Висота, барк	№	Індекс з табл. Б.1	Частота, Гц	Висота, барк
0	1	46.875	0.463	*9*	20	937.500	8.103	*18*	65	3843.750	17.032
1	2	93.750	0.925	*10*	24	1125.000	9.275	*19*	73	4687.500	18.166
2	3	140.625	1.385	*11*	27	1265.625	10.057	*20*	77	5437.500	19.028
3	5	234.375	2.295	*12*	32	1500.000	11.199	*21*	82	6375.000	19.964
4	7	328.125	3.184	*13*	37	1734.375	12.170	*22*	89	7687.500	21.052
5	9	421.875	4.045	*14*	42	1968.750	13.002	*23*	97	9375.000	22.113
6	12	562.500	5.272	*15*	49	2343.750	14.111	*24*	103	11625.000	23.072
7	14	656.250	6.041	*16*	53	2718.750	15.018	*25*	113	15375.000	23.991
8	17	796.875	7.119	*17*	59	3281.250	16.124	*26*	126	20250.000	24.597

Ширина критичних смужок змінюється відносно центральної частоти від майже 0.1 кГц на НЧ до майже 4 кГц на ВЧ і є відомою з психоакустичних експериментів, стислий результат яких полягає у тому, що ССЛ має кращу роздільну здатність по частоті у нижньому частотному діапазоні, аніж у верхньому. Для визначення, чи може локальний максимум бути тональною компонентою, навколо нього аналізується область (діапазон) частот df, яка для *MPEG Layer II* при $f_д = 48$ кГц визначається стандартом [126,127] наступним чином:

$df = 93.750$ Гц для 0 кГц $< f \le 3$ кГц;

$df = 140.63$ Гц для 3 кГц $< f \le 6$ кГц;

$df = 281.25$ Гц для 6 кГц $< f \le 12$ кГц;

$df = 562.50$ Гц для 12 кГц $< f \le 24$ кГц.

Для отримання переліку спектральних складових X_k, що є тональними (ТК) або нетональними (НТК) компонентами, використовують наступні три операції:

Б.4.1. Пошук і маркування локальних максимумів енергетичного спектра ділянки ІКМ-сигналу: компонента X_k вважається локальним максимумом, якщо вона за величиною перевищує попередню (X_{k-1}) і не менше за наступну (X_{k+1}):

$$X_k > X_{k-1} \quad \text{та} \quad X_k \ge X_{k+1}. \tag{Б.5}$$

Б.4.2. Для формування списку ТК і обчислення рівня звукового тиску, досліджується область частот навколо кожної спектральної компоненти, що є локальним максимумом. Відповідна останньому складова заноситься до списку тональних компонент, якщо у досліджуваній навколо неї області частот вона перевищує будь-яку компоненту з заданого околу, за виключенням двох сусідніх з нею, не менш як на 7 дБ, задовольняючи умову:

$$(X_k - X_{k+j}) \ge 7 \text{ дБ}, \tag{Б.6}$$

де індекс j обирається у відповідності до умов (для *MPEG Layer II*):

$j = -2, +2$ для $2 < k < 63$;

$j = -3, -2, +2, +3$ для $63 \le k < 127$;

$j = -6, \ldots, -2, +2, \ldots, +6$ для $127 \le k < 255$;

$j = -12, \ldots, -2, +2, \ldots, +12$ для $255 \le k \le 500$.

Тобто області розширюються з підвищенням частоти (зі зростанням індексу k спектральної компоненти X_k). Як правило, кількість ТК, що задовольняють умові (Б.6), є відносно невеликою.

Якщо встановлено, що X_k є тональною компонентою, то записуються наступні параметри:

- порядковий номер спектральної складової k;
- рівень звукового тиску (при цьому також враховуються рівні двох сусідніх з обраним в якості ТК локальним максимумом складових):

$$X_{тк_k} = X_{k-1} + X_k + X_{k+1} \text{ дБ}; \qquad (Б.7)$$

- мітка тональності.

Потім рівень усіх спектральних компонент всередині аналізованої області частот прирівнюється до $-\infty$.

Б.4.3. Для формування списку НТК і обчислення їх потужності, з первинного спектра сигналу вибірки виключаються усі тональні й сусідні з ними компоненти, що були вже враховані раніше. Нетональні компоненти визначаються серед тих складових, що залишилися. Спектр останніх ділиться на смуги частот, що дорівнюють критичним смужкам слуху. Границі критичних смужок задаються стандартом [126,127]. Для випадку *MPEG Layer II* при $f_д = 48$ кГц використовується 26 критичних смужок (див. табл. Б2).

Усередині кожної критичної смужки потужність спектральних складових підсумовується, формуючи рівень звукового тиску кожної нової НТК, що відповідає поточній смужці й у подальшому заміщує всі шумоподібні складові в її межах — у вигляді однієї компоненти, розміщеної у геометричному центрі смужки (з урахуванням дискретності спектра Δf).

Записуються наступні параметри:

- порядковий номер k спектральної складової, найближчої до геометричної середини критичної смужки;
- рівень звукового тиску $X_{нтк_k}$ у дБ;
- мітка нетональності.

Б.5. Прорідження спектра тональних і нетональних компонент

Процедура прорідження (або так званої децимації) використовується для отримання лише значущих (релевантних) маскуючих сигналів — шляхом зменшення кількості маскуючих компонент, яку необхідно врахувати при подальшому обчисленні глобального порогу маскування.

Б.5.1. Тональні ($X_{тк}$) і нетональні ($X_{нтк}$) компоненти братимуть участь у подальшому обчисленні порогу маскування лише у тому

випадку, якщо вони лежать вище за абсолютний поріг чутності LTq (див. (Б.4)):

$$X_{mk_k} \geq LTq_k \quad \text{або} \quad X_{ntk_k} \geq LTq_k , \qquad (Б.8)$$

де LTq_k — абсолютний поріг (поріг у тиші) на k-й частоті (див. табл. Б.1).

Б.5.2. Крім того, ТК ще додатково проріджуються за допомогою вікна шириною 0,5 барк. Якщо у вікно потрапило дві або більше ТК, то ті з них, що мають менший рівень, зі списку тональних виключаються.

У подальшому, для позначення релевантних тональних (dX_{mk}) і нетональних (dX_{ntk}) маскуючих компонент з отриманого після проріджения остаточного списку використовується індекс j.

Б.6. Обчислення індивідуальних порогів маскування

При обчисленні глобального порогу маскування з початкових $N/2 = 512$ спектральних складових, індексованих через k, може бути взятою до уваги тільки підмножина компонент, проіндексованих через i. Використовувані компоненти залежать від частоти дискретизації і рівня стиснення (див. табл. Б.1).

Після проріджения формується нова сітка спектральних компонент, що у випадку *MPEG Layer II* виглядає наступним чином:

- у перших 3-х субсмугах ($f = 0...2250$ Гц; $k = 1...48$; $i = 1...48$) враховуються всі спектральні компоненти (субвибірка не проводиться);
- у наступних 3-х субсмугах ($f = 2250...4500$ Гц; $k = 48...96$; $i = 48...72$) враховується кожна друга спектральна компонента;
- у подальших 6-и субсмугах ($f = 4500...9000$ Гц; $k = 96...192$; $i = 72...96$) враховується вже кожна четверта спектральна компонента;
- в останніх 20-ти субсмугах ($f = 9000...20250$ Гц; $k = 192...432(512)$; $i = 96...126$) враховується лише кожна восьма спектральна компонента.

Приміром, якщо верхня частота спектру сигналу обмежена значенням 24 кГц (тобто $f_д = 48$ кГц), то після вищенаведеного проріджения ми одержимо спектр, що міститиме загалом 126 компонент.

Кожній ТК і НТК ставиться у відповідність індекс i з субвибіркової області частот, який за частотою є найближчим до первинної спектральної компоненти X_k. Для *MPEG Layer II* при $f_д = 48$ кГц індекс i приймає значення від 0 до 125 (у випадку віднульового індексування).

Індивідуальний поріг маскування тональних і нетональних компонент визначається наступним чином:

$$LT_{тк}[z(j), z(u)] = dX_{тк}[z(j)] + av_{тк}[z(j)] + vf[z(j), z(u)] \text{ дБ}; \quad (Б.9)$$

$$LT_{нтк}[z(j), z(u)] = dX_{нтк}[z(j)] + av_{нтк}[z(j)] + vf[z(j), z(u)] \text{ дБ}, \quad (Б.10)$$

де $LT_{тк}$ і $LT_{нтк}$ — індивідуальні пороги маскування висот тону, відповідно, тональної і нетональної компонент на фоні висот тону критичних смужок *CBR*, заданих стандартом (див. табл. Б.1), — можуть бути як додатною, так і від'ємною величиною; $dX_{*к}[z(j)]$ — складова, що враховує рівень звукового тиску j-ї маскуючої компоненти, якій відповідає висота тону $z(j)$; $av_{*к}[z(j)]$ — показник маскування маскуючої компоненти $dX_{*к}[z(j)]$, який є різним для тональних і нетональних компонент:

- для ТК:

$$av_{тк} = -1{,}525 - 0{,}275 \cdot z(j) - 4{,}5 \text{ дБ}; \quad (Б.11)$$

- для НТК:

$$av_{нтк} = -1{,}525 - 0{,}175 \cdot z(j) - 0{,}5 \text{ дБ}; \quad (Б.11)$$

vf — маскуюча функція компоненти $dX_{*к}[z(j)]$, яка охарактеризується різною крутизною нижньої і верхньої ділянок своєї характеристики, що залежить від відстані dz (у барках) до маскуючої компоненти:

$$dz = z(u) - z(j), \quad (Б.12)$$

де u — індекс спектральної складової, відносно якої обчислюється маскуюча функція; j — індекс маскуючої тональної або нетональної компоненти.

Маскуюча функція vf є однаковою для ТК і НТК і визначається з наступної системи умов:

$$\begin{cases} vf = 17 \cdot (dz + 1) - \{0.4 \cdot X[z(j)] + 6\} \text{ дБ для } -3 \le dz < -1 \text{ барк}; \\ vf = \{0.4 \cdot X[z(j)] + 6\} \cdot dz \text{ дБ для } -1 \le dz < 0 \text{ барк}; \\ vf = -17 \cdot dz \text{ дБ для } 0 \le dz < 1 \text{ барк}; \\ vf = (dz - 1) - \{17 - 0.15 \cdot X[z(j)]\} - 17 \text{ дБ для } 1 \le dz < 8 \text{ барк}. \end{cases} \quad (Б.13)$$

Якщо $dz < -3$ барк або $dz \ge 8$ барк, то маскування далі не розглядається (за межами цього діапазону маскуюча функція має нескінченне загасання і до відповідних елементів $LT_{тк}$ або $LT_{нтк}$ записується значення $-\infty$ — тобто відповідна компонента не здійснюватиме маскування за межами діапазону).

Б.7. Обчислення глобального порогу маскування

Глобальний поріг маскування $LTg(u)$ на u-му частотному відлікові визначається крутизною нижньої і верхньої ділянок індивідуального порогу маскування кожної j-ї тональної і нетональної маскуючої компоненти, а також значенням абсолютного порогу (у тиші) $LTq(u)$ (див. табл. Б.1 і вираз (Б.4)).

Глобальний поріг маскування знаходиться підсумовуванням потужностей, що відповідають індивідуальним порогам маскування LT_{mk} і LT_{ntk}, а також порогу у тиші:

$$LTg(u) = 10 \cdot lg\left\{10^{\frac{LTq(u)}{10}} + \sum_{j=0}^{N_{mk}-1} 10^{\frac{LT_{mk}[z(j),\, z(u)]}{10}} + \sum_{j=0}^{N_{ntk}-1} 10^{\frac{LT_{ntk}[z(j),\, z(u)]}{10}}\right\} \text{ дБ} \quad (Б.14)$$

де N_{mk} і N_{ntk} — загальна кількість ТК і НТК відповідно.

Для заданого u діапазон значень j може бути звужений, охоплюючи лише ті маскуючі компоненти, що знаходяться в межах від –3 до 8 барків відносно u. Поза вказаним діапазоном LT_{mk} і LT_{ntk} дорівнюють $-\infty$.

Б.8. Визначення мінімального порогу маскування

Мінімальний поріг маскування $LTmin(s)$ у субсмузі s визначається як

$$LTmin(s) = \min_{\substack{f(u) \\ \text{у субсмузі } s}} \left[LTg(u)\right] \text{ дБ}, \quad (Б.15)$$

де $f(u)$ — частота u-го частотного відліку, яка є стандартно табульованою (див. табл. Б.1). Мінімальний поріг маскування $LTmin(s)$ обчислюється для кожної з 32-х субсмуг кожного аудіокадру.

Б.9. Обчислення відношення рівня сигналу до порогу маскування

Відношення рівня сигналу до мінімального порогу маскування визначається для кожної субсмуги аудіокадру:

$$SMR(s) = L_{cc}(s) - LTmin(s) \text{ дБ}, \quad (Б.16)$$

де $L_{cc}(s)$ — рівень звукового тиску (енергія сигналу відліку) в s-й субсмузі кодування (див. (Б.3)).

Додаток В.
ОБЧИСЛЕННЯ КОНТРОЛЬНИХ СУМ

Здійснення контролю цілісності передаваних аудіо- та інших інформаційних даних циклічним надмірним кодом *CRC* (*Cyclic Redundancy Check*) дозволяє приймальній стороні виявляти помилки, що виникли під час передавання. З цією метою до складу кадрів і блоків передаваних даних включаються слова *CRC* (контрольні суми), що визначаються за результатом виконання процедури, опис якої подано нижче.

CRC код визначається через генеруючий поліном степеня n:

$$G_n(x) = x^n + g_{n-1} \cdot x^{n-1} + g_{n-2} \cdot x^{n-2} + \ldots + g_2 \cdot x^2 + g_1 \cdot x + 1 \qquad \text{(B.1)}$$

при $n \geq 1$ та $g_i \in \{0,1\}$, $i = 1...(n-1)$.

Обчислення *CRC* може бути здійснене за допомогою n-розрядного (у відповідності до степеня генеруючого полінома) регістра зсуву (рис. В.1).

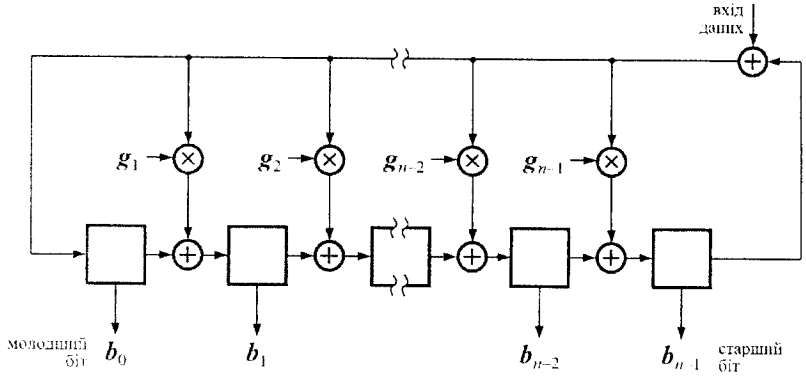


Рис. В.1. Узагальнена блок-схема *CRC*-регістра зсуву

Розряди регістра позначені через $b_0, b_1, \ldots, b_{n-1}$, де b_0 відповідає 1, b_1 — x, b_2 — x^2, ..., b_{n-1} — x^{n-1}. Відведення з регістра зсуву здійснюються за допомогою суматорів по модулю 2 (*XOR*) на входах тих розрядів, що відповідають одиничним коефіцієнтам g_i полінома $G_n(x)$.

На початку обчислення *CRC* вміст усіх розрядів відповідає первинному стану, обумовленому призначенням регістра (усі розряди або

обнуляються, або до них записуються одиниці). Після надходження першого біта блоку даних (найстарший біт йде першим) на вхід регістра, тактовий генератор зсуву (на рис. В.1 не показаний) змушує регістр зсунути вміст своїх розрядів на одну позицію у бік старших розрядів, заповнюючи при цьому розряди відведень результатами відповідних операцій *XOR*. Процедура повторюється для кожного наступного біта даних на вході регістра. По завершенні зсуву після надходження на вхід останнього (наймолодшого) біта з блоку даних, зсувний регістр містить на своїх відведеннях слово *CRC*, яке може бути зчитане і включене до складу структури передаваних даних. Як і вхідні дані регістра, виведення слова *CRC* починається з найстаршого біта.

Використовувані у системах цифрового мовлення *CRC*-коди зазвичай базуються на наведених нижче поліномах:

$$G_{16}(x) = x^{16} + x^{12} + x^5 + 1; \qquad \text{(В.2)}$$

$$G_{16}(x) = x^{16} + x^{15} + x^2 + 1; \qquad \text{(В.3)}$$

$$G_8(x) = x^8 + x^4 + x^3 + x^2 + 1. \qquad \text{(В.4)}$$

Додаток Г.

КОЕФІЦІЄНТИ АНАЛІЗУЮЧОГО І СИНТЕЗУЮЧОГО ВІКОН БЛОКІВ ФІЛЬТРАЦІЇ *MPEG*

Таблиця Г

Коефіцієнти аналізуючого вікна у складі блоку фільтрів аналізу *MPEG*

ζ[0]=+0.000000000	ζ[1]=-0.000000477	ζ[2]=-0.000000477
ζ[3]=-0.000000477	ζ[4]=-0.000000477	ζ[5]=-0.000000477
ζ[6]=-0.000000477	ζ[7]=-0.000000954	ζ[8]=-0.000000954
ζ[9]=-0.000000954	ζ[10]=-0.000000954	ζ[11]=-0.000001431
ζ[12]=-0.000001431	ζ[13]=-0.000001907	ζ[14]=-0.000001907
ζ[15]=-0.000002384	ζ[16]=-0.000002384	ζ[17]=-0.000002861
ζ[18]=-0.000003338	ζ[19]=-0.000003338	ζ[20]=-0.000003815
ζ[21]=-0.000004292	ζ[22]=-0.000004768	ζ[23]=-0.000005245
ζ[24]=-0.000006199	ζ[25]=-0.000006676	ζ[26]=-0.000007629
ζ[27]=-0.000008106	ζ[28]=-0.000009060	ζ[29]=-0.000010014
ζ[30]=-0.000011444	ζ[31]=-0.000012398	ζ[32]=-0.000013828
ζ[33]=-0.000014782	ζ[34]=-0.000016689	ζ[35]=-0.000018120
ζ[36]=-0.000019550	ζ[37]=-0.000021458	ζ[38]=-0.000023365
ζ[39]=-0.000025272	ζ[40]=-0.000027657	ζ[41]=-0.000030041
ζ[42]=-0.000032425	ζ[43]=-0.000034809	ζ[44]=-0.000037670
ζ[45]=-0.000040531	ζ[46]=-0.000043392	ζ[47]=-0.000046253
ζ[48]=-0.000049591	ζ[49]=-0.000052929	ζ[50]=-0.000055790
ζ[51]=-0.000059605	ζ[52]=-0.000062943	ζ[53]=-0.000066280
ζ[54]=-0.000070095	ζ[55]=-0.000073433	ζ[56]=-0.000076771
ζ[57]=-0.000080585	ζ[58]=-0.000083923	ζ[59]=-0.000087261
ζ[60]=-0.000090599	ζ[61]=-0.000093460	ζ[62]=-0.000096321
ζ[63]=-0.000099182	ζ[64]=+0.000101566	ζ[65]=+0.000103951
ζ[66]=+0.000105858	ζ[67]=+0.000107288	ζ[68]=+0.000108242
ζ[69]=+0.000108719	ζ[70]=+0.000108719	ζ[71]=+0.000108242
ζ[72]=+0.000106812	ζ[73]=+0.000105381	ζ[74]=+0.000102520
ζ[75]=+0.000099182	ζ[76]=+0.000095367	ζ[77]=+0.000090122
ζ[78]=+0.000084400	ζ[79]=+0.000077724	ζ[80]=+0.000069618
ζ[81]=+0.000060558	ζ[82]=+0.000050545	ζ[83]=+0.000039577
ζ[84]=+0.000027180	ζ[85]=+0.000013828	ζ[86]=-0.000000954
ζ[87]=-0.000017166	ζ[88]=-0.000034332	ζ[89]=-0.000052929
ζ[90]=-0.000072956	ζ[91]=-0.000093937	ζ[92]=-0.000116348
ζ[93]=-0.000140190	ζ[94]=-0.000165462	ζ[95]=-0.000191212
ζ[96]=-0.000218868	ζ[97]=-0.000247478	ζ[98]=-0.000277042
ζ[99]=-0.000307560	ζ[100]=-0.000339031	ζ[101]=-0.000371456
ζ[102]=-0.000404358	ζ[103]=-0.000438213	ζ[104]=-0.000472546
ζ[105]=-0.000507355	ζ[106]=-0.000542164	ζ[107]=-0.000576973
ζ[108]=-0.000611782	ζ[109]=-0.000646591	ζ[110]=-0.000680923
ζ[111]=-0.000714302	ζ[112]=-0.000747204	ζ[113]=-0.000779152
ζ[114]=-0.000809669	ζ[115]=-0.000838757	ζ[116]=-0.000866413
ζ[117]=-0.000891685	ζ[118]=-0.000915051	ζ[119]=-0.000935555
ζ[120]=-0.000954151	ζ[121]=-0.000968933	ζ[122]=-0.000980854
ζ[123]=-0.000989437	ζ[124]=-0.000994205	ζ[125]=-0.000995159
ζ[126]=-0.000991821	ζ[127]=-0.000983715	ζ[128]=+0.000971317
ζ[129]=+0.000953674	ζ[130]=+0.000930786	ζ[131]=+0.000902653
ζ[132]=+0.000868797	ζ[133]=+0.000829220	ζ[134]=+0.000783920
ζ[135]=+0.000731945	ζ[136]=+0.000674248	ζ[137]=+0.000610352
ζ[138]=+0.000539303	ζ[139]=+0.000462532	ζ[140]=+0.000378609
ζ[141]=+0.000288486	ζ[142]=+0.000191689	ζ[143]=+0.000088215
ζ[144]=-0.000021458	ζ[145]=-0.000137329	ζ[146]=-0.000259876
ζ[147]=-0.000388145	ζ[148]=-0.000522137	ζ[149]=-0.000661850
ζ[150]=-0.000806808	ζ[151]=-0.000956535	ζ[152]=-0.001111031
ζ[153]=-0.001269817	ζ[154]=-0.001432419	ζ[155]=-0.001597881
ζ[156]=-0.001766682	ζ[157]=-0.001937389	ζ[158]=-0.002110004
ζ[159]=-0.002283096	ζ[160]=-0.002457142	ζ[161]=-0.002630711

ζ[162]=-0.002803326	ζ[163]=-0.002974033	ζ[164]=-0.003141880
ζ[165]=-0.003306866	ζ[166]=-0.003467083	ζ[167]=-0.003622532
ζ[168]=-0.003771782	ζ[169]=-0.003914356	ζ[170]=-0.004048824
ζ[171]=-0.004174709	ζ[172]=-0.004290581	ζ[173]=-0.004395962
ζ[174]=-0.004489899	ζ[175]=-0.004570484	ζ[176]=-0.004638195
ζ[177]=-0.004691124	ζ[178]=-0.004728317	ζ[179]=-0.004748821
ζ[180]=-0.004752159	ζ[181]=-0.004737377	ζ[182]=-0.004703045
ζ[183]=-0.004649162	ζ[184]=-0.004573822	ζ[185]=-0.004477024
ζ[186]=-0.004357815	ζ[187]=-0.004215240	ζ[188]=-0.004049301
ζ[189]=-0.003858566	ζ[190]=-0.003643036	ζ[191]=-0.003401756
ζ[192]=+0.003134727	ζ[193]=+0.002841473	ζ[194]=+0.002521515
ζ[195]=+0.002174854	ζ[196]=+0.001800537	ζ[197]=+0.001399517
ζ[198]=+0.000971317	ζ[199]=+0.000515938	ζ[200]=+0.000033379
ζ[201]=-0.000475883	ζ[202]=-0.001011848	ζ[203]=-0.001573563
ζ[204]=-0.002161503	ζ[205]=-0.002774239	ζ[206]=-0.003411293
ζ[207]=-0.004072189	ζ[208]=-0.004756451	ζ[209]=-0.005462170
ζ[210]=-0.006189346	ζ[211]=-0.006937027	ζ[212]=-0.007703304
ζ[213]=-0.008487225	ζ[214]=-0.009287834	ζ[215]=-0.010103703
ζ[216]=-0.010933399	ζ[217]=-0.011775017	ζ[218]=-0.012627602
ζ[219]=-0.013489246	ζ[220]=-0.014358521	ζ[221]=-0.015233517
ζ[222]=-0.016112804	ζ[223]=-0.016994476	ζ[224]=-0.017876148
ζ[225]=-0.018756866	ζ[226]=-0.019634247	ζ[227]=-0.020506859
ζ[228]=-0.021372318	ζ[229]=-0.022228718	ζ[230]=-0.023074150
ζ[231]=-0.023907185	ζ[232]=-0.024725437	ζ[233]=-0.025527000
ζ[234]=-0.026310921	ζ[235]=-0.027073860	ζ[236]=-0.027815342
ζ[237]=-0.028532982	ζ[238]=-0.029224873	ζ[239]=-0.029890060
ζ[240]=-0.030526638	ζ[241]=-0.031132698	ζ[242]=-0.031706810
ζ[243]=-0.032248020	ζ[244]=-0.032754898	ζ[245]=-0.033225536
ζ[246]=-0.033659935	ζ[247]=-0.034055710	ζ[248]=-0.034412861
ζ[249]=-0.034730434	ζ[250]=-0.035007000	ζ[251]=-0.035242081
ζ[252]=-0.035435200	ζ[253]=-0.035586357	ζ[254]=-0.035694122
ζ[255]=-0.035758972	ζ[256]=+0.035780907	ζ[257]=+0.035758972
ζ[258]=+0.035694122	ζ[259]=+0.035586357	ζ[260]=+0.035435200
ζ[261]=+0.035242081	ζ[262]=+0.035007000	ζ[263]=+0.034730434
ζ[264]=+0.034412861	ζ[265]=+0.034055710	ζ[266]=+0.033659935
ζ[267]=+0.033225536	ζ[268]=+0.032754898	ζ[269]=+0.032248020
ζ[270]=+0.031706810	ζ[271]=+0.031132698	ζ[272]=+0.030526638
ζ[273]=+0.029890060	ζ[274]=+0.029224873	ζ[275]=+0.028532982
ζ[276]=+0.027815342	ζ[277]=+0.027073860	ζ[278]=+0.026310921
ζ[279]=+0.025527000	ζ[280]=+0.024725437	ζ[281]=+0.023907185
ζ[282]=+0.023074150	ζ[283]=+0.022228718	ζ[284]=+0.021372318
ζ[285]=+0.020506859	ζ[286]=+0.019634247	ζ[287]=+0.018756866
ζ[288]=+0.017876148	ζ[289]=+0.016994476	ζ[290]=+0.016112804
ζ[291]=+0.015233517	ζ[292]=+0.014358521	ζ[293]=+0.013489246
ζ[294]=+0.012627602	ζ[295]=+0.011775017	ζ[296]=+0.010933399
ζ[297]=+0.010103703	ζ[298]=+0.009287834	ζ[299]=+0.008487225
ζ[300]=+0.007703304	ζ[301]=+0.006937027	ζ[302]=+0.006189346
ζ[303]=+0.005462170	ζ[304]=+0.004756451	ζ[305]=+0.004072189
ζ[306]=+0.003411293	ζ[307]=+0.002774239	ζ[308]=+0.002161503
ζ[309]=+0.001573563	ζ[310]=+0.001011848	ζ[311]=+0.000475883
ζ[312]=-0.000033379	ζ[313]=-0.000515938	ζ[314]=-0.000971317
ζ[315]=-0.001399517	ζ[316]=-0.001800537	ζ[317]=-0.002174854
ζ[318]=-0.002521515	ζ[319]=-0.002841473	ζ[320]=+0.003134727
ζ[321]=+0.003401756	ζ[322]=+0.003643036	ζ[323]=+0.003858566
ζ[324]=+0.004049301	ζ[325]=+0.004215240	ζ[326]=+0.004357815
ζ[327]=+0.004477024	ζ[328]=+0.004573822	ζ[329]=+0.004649162
ζ[330]=+0.004703045	ζ[331]=+0.004737377	ζ[332]=+0.004752159
ζ[333]=+0.004748821	ζ[334]=+0.004728317	ζ[335]=+0.004691124
ζ[336]=+0.004638195	ζ[337]=+0.004570484	ζ[338]=+0.004489899
ζ[339]=+0.004395962	ζ[340]=+0.004290581	ζ[341]=+0.004174709
ζ[342]=+0.004048824	ζ[343]=+0.003914356	ζ[344]=+0.003771782
ζ[345]=+0.003622532	ζ[346]=+0.003467083	ζ[347]=+0.003306866
ζ[348]=+0.003141880	ζ[349]=+0.002974033	ζ[350]=+0.002803326
ζ[351]=+0.002630711	ζ[352]=+0.002457142	ζ[353]=+0.002283096
ζ[354]=+0.002110004	ζ[355]=+0.001937389	ζ[356]=+0.001766682

ζ[357]=+0.001597881	ζ[358]=+0.001432419	ζ[359]=+0.001269817
ζ[360]=+0.001111031	ζ[361]=+0.000956535	ζ[362]=+0.000806808
ζ[363]=+0.000661850	ζ[364]=+0.000522137	ζ[365]=+0.000388145
ζ[366]=+0.000259876	ζ[367]=+0.000137329	ζ[368]=+0.000021458
ζ[369]=-0.000088215	ζ[370]=-0.000191689	ζ[371]=-0.000288486
ζ[372]=-0.000378609	ζ[373]=-0.000462532	ζ[374]=-0.000539303
ζ[375]=-0.000610352	ζ[376]=-0.000674248	ζ[377]=-0.000731945
ζ[378]=-0.000783920	ζ[379]=-0.000829220	ζ[380]=-0.000868797
ζ[381]=-0.000902653	ζ[382]=-0.000930786	ζ[383]=-0.000953674
ζ[384]=+0.000971317	ζ[385]=+0.000983715	ζ[386]=+0.000991821
ζ[387]=+0.000995159	ζ[388]=+0.000994205	ζ[389]=+0.000989437
ζ[390]=+0.000980854	ζ[391]=+0.000968933	ζ[392]=+0.000954151
ζ[393]=+0.000935555	ζ[394]=+0.000915051	ζ[395]=+0.000891685
ζ[396]=+0.000866413	ζ[397]=+0.000838757	ζ[398]=+0.000809669
ζ[399]=+0.000779152	ζ[400]=+0.000747204	ζ[401]=+0.000714302
ζ[402]=+0.000680923	ζ[403]=+0.000646591	ζ[404]=+0.000611782
ζ[405]=+0.000576973	ζ[406]=+0.000542164	ζ[407]=+0.000507355
ζ[408]=+0.000472546	ζ[409]=+0.000438213	ζ[410]=+0.000404358
ζ[411]=+0.000371456	ζ[412]=+0.000339031	ζ[413]=+0.000307560
ζ[414]=+0.000277042	ζ[415]=+0.000247478	ζ[416]=+0.000218868
ζ[417]=+0.000191212	ζ[418]=+0.000165462	ζ[419]=+0.000140190
ζ[420]=+0.000116348	ζ[421]=+0.000093937	ζ[422]=+0.000072956
ζ[423]=+0.000052929	ζ[424]=+0.000034332	ζ[425]=+0.000017166
ζ[426]=+0.000000954	ζ[427]=-0.000013828	ζ[428]=-0.000027180
ζ[429]=-0.000039577	ζ[430]=-0.000050545	ζ[431]=-0.000060558
ζ[432]=-0.000069618	ζ[433]=-0.000077724	ζ[434]=-0.000084400
ζ[435]=-0.000090122	ζ[436]=-0.000095367	ζ[437]=-0.000099182
ζ[438]=-0.000102520	ζ[439]=-0.000105381	ζ[440]=-0.000106812
ζ[441]=-0.000108242	ζ[442]=-0.000108719	ζ[443]=-0.000108719
ζ[444]=-0.000108242	ζ[445]=-0.000107288	ζ[446]=-0.000105858
ζ[447]=-0.000103951	ζ[448]=+0.000101566	ζ[449]=+0.000099182
ζ[450]=+0.000096321	ζ[451]=+0.000093460	ζ[452]=+0.000090599
ζ[453]=+0.000087261	ζ[454]=+0.000083923	ζ[455]=+0.000080585
ζ[456]=+0.000076771	ζ[457]=+0.000073433	ζ[458]=+0.000070095
ζ[459]=+0.000066280	ζ[460]=+0.000062943	ζ[461]=+0.000059605
ζ[462]=+0.000055790	ζ[463]=+0.000052929	ζ[464]=+0.000049591
ζ[465]=+0.000046253	ζ[466]=+0.000043392	ζ[467]=+0.000040531
ζ[468]=+0.000037670	ζ[469]=+0.000034809	ζ[470]=+0.000032425
ζ[471]=+0.000030041	ζ[472]=+0.000027657	ζ[473]=+0.000025272
ζ[474]=+0.000023365	ζ[475]=+0.000021458	ζ[476]=+0.000019550
ζ[477]=+0.000018120	ζ[478]=+0.000016689	ζ[479]=+0.000014782
ζ[480]=+0.000013828	ζ[481]=+0.000012398	ζ[482]=+0.000011444
ζ[483]=+0.000010014	ζ[484]=+0.000009060	ζ[485]=+0.000008106
ζ[486]=+0.000007629	ζ[487]=+0.000006676	ζ[488]=+0.000006199
ζ[489]=+0.000005245	ζ[490]=+0.000004768	ζ[491]=+0.000004292
ζ[492]=+0.000003815	ζ[493]=+0.000003338	ζ[494]=+0.000003338
ζ[495]=+0.000002861	ζ[496]=+0.000002384	ζ[497]=+0.000002384
ζ[498]=+0.000001907	ζ[499]=+0.000001907	ζ[500]=+0.000001431
ζ[501]=+0.000001431	ζ[502]=+0.000000954	ζ[503]=+0.000000954
ζ[504]=+0.000000954	ζ[505]=+0.000000954	ζ[506]=+0.000000477
ζ[507]=+0.000000477	ζ[508]=+0.000000477	ζ[509]=+0.000000477
ζ[510]=+0.000000477	ζ[511]=+0.000000477	

Таблиця Г.2

Косфіцієнти синтезуючого вікна у складі блоку фільтрів синтезу *MPEG*

ξ[0]=+0.000000000	ξ[1]=-0.000015259	ξ[2]=-0.000015259
ξ[3]=-0.000015259	ξ[4]=-0.000015259	ξ[5]=-0.000015259
ξ[6]=-0.000015259	ξ[7]=-0.000030518	ξ[8]=-0.000030518
ξ[9]=-0.000030518	ξ[10]=-0.000030518	ξ[11]=-0.000045776
ξ[12]=-0.000045776	ξ[13]=-0.000061035	ξ[14]=-0.000061035
ξ[15]=-0.000076294	ξ[16]=-0.000076294	ξ[17]=-0.000091553
ξ[18]=-0.000106812	ξ[19]=-0.000106812	ξ[20]=-0.000122070

ξ[21]=-0.000137329	ξ[22]=-0.000152588	ξ[23]=-0.000167847
ξ[24]=-0.000198364	ξ[25]=-0.000213623	ξ[26]=-0.000244141
ξ[27]=-0.000259399	ξ[28]=-0.000289917	ξ[29]=-0.000320435
ξ[30]=-0.000366211	ξ[31]=-0.000396729	ξ[32]=-0.000442505
ξ[33]=-0.000473022	ξ[34]=-0.000534058	ξ[35]=-0.000579834
ξ[36]=-0.000625610	ξ[37]=-0.000686646	ξ[38]=-0.000747681
ξ[39]=-0.000808716	ξ[40]=-0.000885010	ξ[41]=-0.000961304
ξ[42]=-0.001037598	ξ[43]=-0.001113892	ξ[44]=-0.001205444
ξ[45]=-0.001296997	ξ[46]=-0.001388550	ξ[47]=-0.001480103
ξ[48]=-0.001586914	ξ[49]=-0.001693726	ξ[50]=-0.001785278
ξ[51]=-0.001907349	ξ[52]=-0.002014160	ξ[53]=-0.002120972
ξ[54]=-0.002243042	ξ[55]=-0.002349854	ξ[56]=-0.002456665
ξ[57]=-0.002578735	ξ[58]=-0.002685547	ξ[59]=-0.002792358
ξ[60]=-0.002899170	ξ[61]=-0.002990723	ξ[62]=-0.003082275
ξ[63]=-0.003173828	ξ[64]=+0.003250122	ξ[65]=+0.003326416
ξ[66]=+0.003387451	ξ[67]=+0.003433228	ξ[68]=+0.003463745
ξ[69]=+0.003479004	ξ[70]=+0.003463745	ξ[71]=+0.003463745
ξ[72]=+0.003417969	ξ[73]=+0.003372192	ξ[74]=+0.003280640
ξ[75]=+0.003173828	ξ[76]=+0.003051758	ξ[77]=+0.002883911
ξ[78]=+0.002700806	ξ[79]=+0.002487183	ξ[80]=+0.002227783
ξ[81]=+0.001937866	ξ[82]=+0.001617432	ξ[83]=+0.001266479
ξ[84]=+0.000869751	ξ[85]=+0.000442505	ξ[86]=-0.000030518
ξ[87]=-0.000549316	ξ[88]=-0.001098633	ξ[89]=-0.001693726
ξ[90]=-0.002334595	ξ[91]=-0.003005981	ξ[92]=-0.003723145
ξ[93]=-0.004486084	ξ[94]=-0.005294800	ξ[95]=-0.006118774
ξ[96]=-0.007003784	ξ[97]=-0.007919312	ξ[98]=-0.008865356
ξ[99]=-0.009841919	ξ[100]=-0.010848999	ξ[101]=-0.011886597
ξ[102]=-0.012939453	ξ[103]=-0.014022827	ξ[104]=-0.015121460
ξ[105]=-0.016235352	ξ[106]=-0.017349243	ξ[107]=-0.018463135
ξ[108]=-0.019577026	ξ[109]=-0.020690918	ξ[110]=-0.021789551
ξ[111]=-0.022857666	ξ[112]=-0.023910522	ξ[113]=-0.024932861
ξ[114]=-0.025909424	ξ[115]=-0.026840210	ξ[116]=-0.027725220
ξ[117]=-0.028533936	ξ[118]=-0.029281616	ξ[119]=-0.029937744
ξ[120]=-0.030532837	ξ[121]=-0.031005859	ξ[122]=-0.031387329
ξ[123]=-0.031661987	ξ[124]=-0.031814575	ξ[125]=-0.031845093
ξ[126]=-0.031738281	ξ[127]=-0.031478882	ξ[128]=+0.031082153
ξ[129]=+0.030517578	ξ[130]=+0.029785156	ξ[131]=+0.028884888
ξ[132]=+0.027801514	ξ[133]=+0.026535034	ξ[134]=+0.025085449
ξ[135]=+0.023422241	ξ[136]=+0.021575928	ξ[137]=+0.019531250
ξ[138]=+0.017257690	ξ[139]=+0.014801025	ξ[140]=+0.012115479
ξ[141]=+0.009231567	ξ[142]=+0.006134033	ξ[143]=+0.002822876
ξ[144]=-0.000686646	ξ[145]=-0.004394531	ξ[146]=-0.008316040
ξ[147]=-0.012420654	ξ[148]=-0.016708374	ξ[149]=-0.021179199
ξ[150]=-0.025817871	ξ[151]=-0.030609131	ξ[152]=-0.035552979
ξ[153]=-0.040634155	ξ[154]=-0.045837402	ξ[155]=-0.051132202
ξ[156]=-0.056533813	ξ[157]=-0.061996460	ξ[158]=-0.067520142
ξ[159]=-0.073059082	ξ[160]=-0.078628540	ξ[161]=-0.084182739
ξ[162]=-0.089706421	ξ[163]=-0.095169067	ξ[164]=-0.100540161
ξ[165]=-0.105819702	ξ[166]=-0.110946655	ξ[167]=-0.115921021
ξ[168]=-0.120697021	ξ[169]=-0.125259399	ξ[170]=-0.129562378
ξ[171]=-0.133590698	ξ[172]=-0.137298584	ξ[173]=-0.140670776
ξ[174]=-0.143676758	ξ[175]=-0.146255493	ξ[176]=-0.148422241
ξ[177]=-0.150115967	ξ[178]=-0.151306152	ξ[179]=-0.151962280
ξ[180]=-0.152069092	ξ[181]=-0.151596069	ξ[182]=-0.150497437
ξ[183]=-0.148773193	ξ[184]=-0.146362305	ξ[185]=-0.143264771
ξ[186]=-0.139450073	ξ[187]=-0.134887695	ξ[188]=-0.129577637
ξ[189]=-0.123474121	ξ[190]=-0.116577148	ξ[191]=-0.108856201
ξ[192]=+0.100311279	ξ[193]=+0.090927124	ξ[194]=+0.080688477
ξ[195]=+0.069595337	ξ[196]=+0.057617187	ξ[197]=+0.044784546
ξ[198]=+0.031082153	ξ[199]=+0.016510010	ξ[200]=+0.001068115
ξ[201]=-0.015228271	ξ[202]=-0.032379150	ξ[203]=-0.050354004
ξ[204]=-0.069168091	ξ[205]=-0.088775635	ξ[206]=-0.109161377
ξ[207]=-0.130310059	ξ[208]=-0.152206421	ξ[209]=-0.174789429
ξ[210]=-0.198059082	ξ[211]=-0.221984863	ξ[212]=-0.246505737
ξ[213]=-0.271591187	ξ[214]=-0.297210693	ξ[215]=-0.323318481

ξ[216]=-0.349868774	ξ[217]=-0.376800537	ξ[218]=-0.404083252
ξ[219]=-0.431655884	ξ[220]=-0.459472656	ξ[221]=-0.487472534
ξ[222]=-0.515609741	ξ[223]=-0.543823242	ξ[224]=-0.572036743
ξ[225]=-0.600219727	ξ[226]=-0.628295898	ξ[227]=-0.656219482
ξ[228]=-0.683914185	ξ[229]=-0.711318970	ξ[230]=-0.738372803
ξ[231]=-0.765029907	ξ[232]=-0.791213989	ξ[233]=-0.816864014
ξ[234]=-0.841949463	ξ[235]=-0.866363525	ξ[236]=-0.890090942
ξ[237]=-0.913055420	ξ[238]=-0.935195923	ξ[239]=-0.956481934
ξ[240]=-0.976852417	ξ[241]=-0.996246338	ξ[242]=-1.014617920
ξ[243]=-1.031936646	ξ[244]=-1.048156738	ξ[245]=-1.063217163
ξ[246]=-1.077117920	ξ[247]=-1.089782715	ξ[248]=-1.101211548
ξ[249]=-1.111373901	ξ[250]=-1.120223999	ξ[251]=-1.127746582
ξ[252]=-1.133926392	ξ[253]=-1.138763428	ξ[254]=-1.142211914
ξ[255]=-1.144287109	ξ[256]= 1.144989014	ξ[257]= 1.144287109
ξ[258]= 1.142211914	ξ[259]= 1.138763428	ξ[260]= 1.133926392
ξ[261]= 1.127746582	ξ[262]= 1.120223999	ξ[263]= 1.111373901
ξ[264]= 1.101211548	ξ[265]= 1.089782715	ξ[266]= 1.077117920
ξ[267]= 1.063217163	ξ[268]= 1.048156738	ξ[269]= 1.031936646
ξ[270]= 1.014617920	ξ[271]=+0.996246338	ξ[272]=+0.976852417
ξ[273]=+0.956481934	ξ[274]=+0.935195923	ξ[275]=+0.913055420
ξ[276]=+0.890090942	ξ[277]=+0.866363525	ξ[278]=+0.841949463
ξ[279]=+0.816864014	ξ[280]=+0.791213989	ξ[281]=+0.765029907
ξ[282]=+0.738372803	ξ[283]=+0.711318970	ξ[284]=+0.683914185
ξ[285]=+0.656219482	ξ[286]=+0.628295898	ξ[287]=+0.600219727
ξ[288]=+0.572036743	ξ[289]=+0.543823242	ξ[290]=+0.515609741
ξ[291]=+0.487472534	ξ[292]=+0.459472656	ξ[293]=+0.431655884
ξ[294]=+0.404083252	ξ[295]=+0.376800537	ξ[296]=+0.349868774
ξ[297]=+0.323318481	ξ[298]=+0.297210693	ξ[299]=+0.271591187
ξ[300]=+0.246505737	ξ[301]=+0.221984863	ξ[302]=+0.198059082
ξ[303]=+0.174789429	ξ[304]=+0.152206421	ξ[305]=+0.130310059
ξ[306]=+0.109161377	ξ[307]=+0.088775635	ξ[308]=+0.069168091
ξ[309]=+0.050354004	ξ[310]=+0.032379150	ξ[311]=+0.015228271
ξ[312]=-0.001068115	ξ[313]=-0.016510010	ξ[314]=-0.031082153
ξ[315]=-0.044784546	ξ[316]=-0.057617187	ξ[317]=-0.069595337
ξ[318]=-0.080688477	ξ[319]=-0.090927124	ξ[320]=+0.100311279
ξ[321]=+0.108856201	ξ[322]=+0.116577148	ξ[323]=+0.123474121
ξ[324]=+0.129577637	ξ[325]=+0.134887695	ξ[326]=+0.139450073
ξ[327]=+0.143264771	ξ[328]=+0.146362305	ξ[329]=+0.148773193
ξ[330]=+0.150497437	ξ[331]=+0.151596069	ξ[332]=+0.152069092
ξ[333]=+0.151962280	ξ[334]=+0.151306152	ξ[335]=+0.150115967
ξ[336]=+0.148422241	ξ[337]=+0.146255493	ξ[338]=+0.143676758
ξ[339]=+0.140670776	ξ[340]=+0.137298584	ξ[341]=+0.133590698
ξ[342]=+0.129562378	ξ[343]=+0.125259399	ξ[344]=+0.120697021
ξ[345]=+0.115921021	ξ[346]=+0.110946655	ξ[347]=+0.105819702
ξ[348]=+0.100540161	ξ[349]=+0.095169067	ξ[350]=+0.089706421
ξ[351]=+0.084182739	ξ[352]=+0.078628540	ξ[353]=+0.073059082
ξ[354]=+0.067520142	ξ[355]=+0.061996460	ξ[356]=+0.056533813
ξ[357]=+0.051132202	ξ[358]=+0.045837402	ξ[359]=+0.040634155
ξ[360]=+0.035552979	ξ[361]=+0.030609131	ξ[362]=+0.025817871
ξ[363]=+0.021179199	ξ[364]=+0.016708374	ξ[365]=+0.012420654
ξ[366]=+0.008316040	ξ[367]=+0.004394531	ξ[368]=+0.000686646
ξ[369]=-0.002822876	ξ[370]=-0.006134033	ξ[371]=-0.009231567
ξ[372]=-0.012115479	ξ[373]=-0.014801025	ξ[374]=-0.017257690
ξ[375]=-0.019531250	ξ[376]=-0.021575928	ξ[377]=-0.023422241
ξ[378]=-0.025085449	ξ[379]=-0.026535034	ξ[380]=-0.027801514
ξ[381]=-0.028884888	ξ[382]=-0.029785156	ξ[383]=-0.030517578
ξ[384]=+0.031082153	ξ[385]=+0.031478882	ξ[386]=+0.031738281
ξ[387]=+0.031845093	ξ[388]=+0.031814575	ξ[389]=+0.031661987
ξ[390]=+0.031387329	ξ[391]=+0.031005859	ξ[392]=+0.030532837
ξ[393]=+0.029937744	ξ[394]=+0.029281616	ξ[395]=+0.028533936
ξ[396]=+0.027725220	ξ[397]=+0.026840210	ξ[398]=+0.025909424
ξ[399]=+0.024932861	ξ[400]=+0.023910522	ξ[401]=+0.022857666
ξ[402]=+0.021789551	ξ[403]=+0.020690918	ξ[404]=+0.019577026
ξ[405]=+0.018463135	ξ[406]=+0.017349243	ξ[407]=+0.016235352
ξ[408]=+0.015121460	ξ[409]=+0.014022827	ξ[410]=+0.012939453

ξ[411]=+0.011886597	ξ[412]=+0.010848999	ξ[413]=+0.009841919
ξ[414]=+0.008865356	ξ[415]=+0.007919312	ξ[416]=+0.007003784
ξ[417]=+0.006118774	ξ[418]=+0.005294800	ξ[419]=+0.004486084
ξ[420]=+0.003723145	ξ[421]=+0.003005981	ξ[422]=+0.002334595
ξ[423]=+0.001693726	ξ[424]=+0.001098633	ξ[425]=+0.000549316
ξ[426]=+0.000030518	ξ[427]=-0.000442505	ξ[428]=-0.000869751
ξ[429]=-0.001266479	ξ[430]=-0.001617432	ξ[431]=-0.001937866
ξ[432]=-0.002227783	ξ[433]=-0.002487183	ξ[434]=-0.002700806
ξ[435]=-0.002883911	ξ[436]=-0.003051758	ξ[437]=-0.003173828
ξ[438]=-0.003280640	ξ[439]=-0.003372192	ξ[440]=-0.003417969
ξ[441]=-0.003463745	ξ[442]=-0.003479004	ξ[443]=-0.003479004
ξ[444]=-0.003463745	ξ[445]=-0.003433228	ξ[446]=-0.003387451
ξ[447]=-0.003326416	ξ[448]=+0.003250122	ξ[449]=+0.003173828
ξ[450]=+0.003082275	ξ[451]=+0.002990723	ξ[452]=+0.002899170
ξ[453]=+0.002792358	ξ[454]=+0.002685547	ξ[455]=+0.002578735
ξ[456]=+0.002456665	ξ[457]=+0.002349854	ξ[458]=+0.002243042
ξ[459]=+0.002120972	ξ[460]=+0.002014160	ξ[461]=+0.001907349
ξ[462]=+0.001785278	ξ[463]=+0.001693726	ξ[464]=+0.001586914
ξ[465]=+0.001480103	ξ[466]=+0.001388550	ξ[467]=+0.001296997
ξ[468]=+0.001205444	ξ[469]=+0.001113892	ξ[470]=+0.001037598
ξ[471]=+0.000961304	ξ[472]=+0.000885010	ξ[473]=+0.000808716
ξ[474]=+0.000747681	ξ[475]=+0.000686646	ξ[476]=+0.000625610
ξ[477]=+0.000579834	ξ[478]=+0.000534058	ξ[479]=+0.000473022
ξ[480]=+0.000442505	ξ[481]=+0.000396729	ξ[482]=+0.000366211
ξ[483]=+0.000320435	ξ[484]=+0.000289917	ξ[485]=+0.000259399
ξ[486]=+0.000244141	ξ[487]=+0.000213623	ξ[488]=+0.000198364
ξ[489]=+0.000167847	ξ[490]=+0.000152588	ξ[491]=+0.000137329
ξ[492]=+0.000122070	ξ[493]=+0.000106812	ξ[494]=+0.000106812
ξ[495]=+0.000091553	ξ[496]=+0.000076294	ξ[497]=+0.000076294
ξ[498]=+0.000061035	ξ[499]=+0.000061035	ξ[500]=+0.000045776
ξ[501]=+0.000045776	ξ[502]=+0.000030518	ξ[503]=+0.000030518
ξ[504]=+0.000030518	ξ[505]=+0.000030518	ξ[506]=+0.000015259
ξ[507]=+0.000015259	ξ[508]=+0.000015259	ξ[509]=+0.000015259
ξ[510]=+0.000015259	ξ[511]=+0.000015259	

Додаток Д.
ВБУДОВАНІ ОПЕРАТОРИ *MATHCAD*

У наведеному нижче переліку операторів використовуються наступні умовні позначення:

- **A** та **B** — масиви (вектори або матриці);
- **u** та **v** — вектори з дійсними або комплексними елементами;
- **M** — квадратна матриця;
- **z** і **w** — дійсні або комплексні числа;
- **x** та **y** — дійсні числа;
- **m** і **n** — цілі числа;
- **i** — діапазон змінних (дискретний аргумент);
- **t** — будь-яке ім'я змінної;
- **f** — будь-яка функція;
- **X** та **Y** — змінні або вирази довільного типу.

На їхнє місце можуть підставлятися об'єкти відповідного типу (змінні, вектори, матриці і т. п.) з будь-якими іншими іменами, а також вирази з відповідного типу результатом їх обчислення.

Оператор	*Позначення*	*Введення*	*Призначення оператора (операції)*
Присвоєння значення напівглобальній змінній (елементу масиву, стовпцю матриці), визначення функції користувача	■ := ■ **z := x + y** $B_{n,m}$:= 8 $A^{<n>}$:= **v** **f(x, [y,...]) := x+7**	:	Присвоює значення змінній (елементу масиву), визначає функцію користувача, видиму правіше і нижче даного оператора
Присвоєння значення глобальній змінній (елементу масиву, стовпцю матриці), визначення функції користувача	■ ≡ ■ **z ≡ x + y** $B_{n,m}$ ≡ 8 $A^{<n>}$ ≡ **v** **f(x, [y,...]) ≡ x+7**	~	Присвоює значення змінній (елементу масиву), визначає функцію користувача, видиму в усьому документі *Mathcad*
Присвоєння значення локальній змінній (елементу масиву, стовпцю матриці)	■ ← ■ **z ← x + y** $B_{n,m}$ ← 8 $A^{<n>}$ ← **v**	{	Присвоює значення локальній змінній (елементу масиву) у межах програмного модуля
Обчислення числового значення	■ = ■·[■] **z** = 1980 $B_{n,m}$ = 8 v_n = 1·s **f(x)** = 74	=	Обчислює й виводить на екран у 2-й операнд числове значення змінної, виразу, функції, записаної у 1-му операнді. Також можна задавати розмірність (3-й операнд)
Круглі дужки	(**X**)	'	Зміна пріоритету виконання операцій, група операторів
Нижній індекс	A_n	[	Задання індексованої змінної
Верхній індекс	$A^{<n>}$	[Ctrl]6	Обрання **n**-го стовпця з масиву **A**

Оператор	*Позначення*	*Введення*	*Призначення оператора (операції)*
Векторизація	$\overrightarrow{f(A)}$	[Ctrl]–	Виконання заданої операції **f** для всіх елементів масиву **A**
Факторіал	**n**!	!	Обчислення факторіалу для цілого додатного числа **n**
Спряжене комплексне число	$\overline{z}$	"	Обчислення спряженого комплексного числа (інвертований сигнал уявної частини **z**)
Транспонування	A^T	[Ctrl]1	Транспонування масиву **A**
Піднесення до степеня	z^w	^	Піднесення **z** до степеня **w**
Степінь матриці, інверсна матриця	M^n	^	Піднесення квадратної матриці **M** до степеня **n** (при **n** = –1 — інверсія матриці)
Заперечення	–**X**	–	Добуток **X** на –1
Сума елементів вектора	Σ**v**	[Ctrl]4	Обчислення суми елементів вектора **v** (повертається скалярне значення)
Квадратний корінь	$\sqrt{w}$	\	Обчислення квадратного кореня з **w**
Корінь **n**-го степеня	$\sqrt[n]{w}$	[Ctrl]\	Обчислення кореня степеня **n** з **w**
Модуль комплексного числа	|**z**|	|	Обчислення результату $\sqrt{Re(z)^2 + Im(z)^2}$
Абсолютна величина вектора (евклідова норма або розмірність вектора)	|**v**|	|	Обчислення $\sqrt{v \cdot v}$, якщо всі елементи **v** є дійсними, і $\sqrt{v \cdot \overline{v}}$, якщо елементи **v** є комплексними
Детермінант матриці **M**	|**M**|	|	Повертає визначник (детермінант) квадратної матриці **M**
Ділення	$\frac{X}{z}$ або **X**÷**z**	/	Ділення виразу **X** на скаляр **z**, що не дорівнює 0 (якщо **X** є масивом, то на **z** ділиться кожен елемент масиву)
Множення	**X**·**Y**	*	Обчислення добутку **X** на **Y**, якщо **X** і **Y** є скалярами. Множення кожного елемента **Y** на **X**, якщо **Y** є масивом, а **X** — скаляром. Обчислення скалярного добутку, якщо **X** та **Y** — вектори однакового розміру. Множення матриць, якщо **X** та **Y** — подібні матриці
Крос-добуток	**u**×**v**	[Ctrl]8	Обчислення векторного добутку векторів **u** та **v**
Підсумовування для скінченного ряду	$\sum_{i=m}^{n} X$	[Ctrl] [Shift]4	Обчислення суми членів **X** для **i** = **m**, **m**+1, ..., **n**, причому **X** може бути будь-яким виразом

Оператор	*Позначення*	*Введення*	*Призначення оператора (операції)*
Добуток для скінченного ряду	$\prod_{i=m}^{n} X$	[Ctrl] [Shift]3	Перемноження елементів **X** для **i = m**, **m+1**,..., **n**, причому **X** може бути будь-яким виразом
Підсумовування для нескінченного ряду	$\sum_{i} X$	$	Обчислення суми нескінченної кількості членів **X**, причому **X** може бути будь-яким виразом
Добуток для нескінченного ряду	$\prod_{i} X$	#	Перемноження нескінченної кількості членів **X**, причому **X** може бути будь-яким виразом
Границя функції в заданій точці	$\lim_{x \to a} f(x)$	[Ctrl]L	Обчислення границі функції **f(x)** при **x**, що прямує до **a** (лише в символьному вигляді)
Границя функції ліворуч від заданої точки	$\lim_{x \to a^-} f(x)$	[Ctrl]B	Обчислення границі функції **f(x)** при **x**, що прямує до **a** зліва (лише в символьному вигляді)
Границя функції праворуч від заданої точки	$\lim_{x \to a^+} f(x)$	[Ctrl]A	Обчислення границі функції **f(x)** при **x**, що прямує до **a** праворуч (лише в символьному вигляді)
Визначений інтеграл	$\int_{a}^{b} f(t)\,dt$	&	Обчислення визначеного інтеграла від підінтегральної функції **f(t)** з межами інтегрування **a** і **b**
Невизначений інтеграл	$\int f(t)\,dt$	[Ctrl]I	Обчислення у символьному вигляді невизначеного інтеграла від підінтегральної функції **f(t)**
Похідна заданої функції по змінній **t**	$\frac{d}{dt} f(t)$	?	Обчислення першої похідної функції **f(t)** по змінній **t**
n-а похідні заданої функції по змінній **t**	$\frac{d^n}{dt^n} f(t)$	[Ctrl]?	Обчислення **n**-ї похідної функції **f(t)** по змінній **t**
Додавання	X+Y	+	Скалярне додавання, якщо **X**, **Y** є скалярами. Додавання елементів, якщо **X** і **Y** — масиви однакового розміру. Якщо **X** є масивом, а **Y** — скаляром, складання кожного елемента **X** з **Y**
Віднімання	X–Y	–	Виконує віднімання скалярів, векторів або матриць **X** і **Y**
Перенесення на інший рядок	X ... +Y	[Ctrl] ↵	Перенесення частини виразу на наступний рядок. Лінія розриву має редакційне значення
Більше за	x > y "T1" > "T2"	>	Повертає 1, якщо **x** > **y**, інакше — 0. **x** та **y** мають бути дійсними скалярними змінними. Для текстових змінних **T1** і **T2** — повертає 1, якщо *ASCII*-код змінної **T1** більше, аніж у **T2**

Оператор	*Позначення*	*Введення*	*Призначення оператора (операції)*
Менше за	**x < y** **"T1" < "T2"**	<	Повертає 1, якщо **x < y**, інакше — 0. **x** та **y** мають бути дійсними скалярними змінними. Для текстових змінних **T1** і **T2** — повертає 1, якщо *ASCII*-код змінної **T1** менше, аніж у **T2**
Більше або дорівнює	**x ≥ y** **"T1" ≥ "T2"**	[Ctrl]0	Повертає 1, якщо **x ≥ y**, 0 — в іншому випадку. **x** та **y** мають бути дійсними скалярними змінними. Для текстових змінних **T1** і **T2** — повертає 1, якщо *ASCII*-код **T1 ≥ T2**
Менше або дорівнює	**x ≤ y** **"T1" ≤ "T2"**	[Ctrl]9	Повертає 1, якщо **x ≤ y**, 0 — в іншому випадку. **x** та **y** мають бути дійсними скалярними змінними. Для текстових змінних **T1** і **T2** — повертає 1, якщо *ASCII*-код **T1 ≤ T2**
Дорівнює	**x = y** **"T1" = "T2"**	[Ctrl]=	Повертає 1, якщо **x = y**, 0 — в іншому випадку. **x** та **y** мають бути дійсними скалярними змінними. Для текстових змінних **T1** і **T2** — повертає 1, якщо *ASCII*-код **T1 = T2**
Не дорівнює	**x ≠ y** **"T1" ≠ "T2"**	[Ctrl]3	Повертає 1, якщо **x ≠ y**, 0 — в іншому випадку. **x** та **y** мають бути дійсними скалярними змінними. Для текстових змінних **T1** і **T2** — повертає 1, якщо *ASCII*-код **T1 ≠ T2**

Додаток Е.
ОСНОВНІ ВБУДОВАНІ ФУНКЦІЇ *MATHCAD*

У наведеному нижче переліку функцій використовуються наступні умовні позначення:

- **A** та **B** — масиви (вектори або матриці);
- **M** і **N** — квадратні матриці;
- **X** та **Y** — змінні або вирази довільного типу;
- **F** — функція-вектор;
- **f** — функція-скаляр;
- **S** — рядкова змінна або константа;
- **x** та **y** — дійсні числа;
- **z** і **w** — дійсні або комплексні числа;
- **m**, **n**, **i**, **j**, **k** — цілі числа;
- **u** та **v**[...] — вектори;
- *file* — ім'я файлу чи файлова змінна, приєднана до імені файлу.

Усі кути в тригонометричних функціях виражені в радіанах. Багатозначні функції та функції з комплексним аргументом завжди повертають головне значення. Імена представлених функцій нечутливі до шрифту, але чутливі до регістра — їх слід вводити в точності, як вони наведені. Всі функції повертають вказане для них значення.

Функції для роботи з комплексними числами:

arg(**z**) — повертає аргумент комплексного числа **z** між $-\pi$ та π;

csgn(**z**) — повертає комплексний знак числа: 0, якщо **z** = 0;
1, якщо **Re**(**z**) > 0 або (**Re**(**z**) = 0 та **Im**(**z**) > 0);
−1 — в іншому випадку;

Im(**z**) — повертає уявну частину комплексного аргументу **z**;

Re(**z**) — повертає дійсну частину комплексного аргументу **z**;

signum(**z**) — повертає комплексний знак числа: 1, якщо **z** = 0;
z/|**z**| — в іншому випадку.

Кусково-неперервні функції:

if(**cond**, **f1**, **f2**) — умовний вираз, що повертає вираз **f1**, якщо логічна умова **cond** виконується, і вираз **f2** в інших випадках;

δ(**m**, **n**) — функція Кронекера: 1, якщо **m** = **n**; 0, якщо **m** ≠ **n**;

Φ(**x**) — одинична східчаста функція (функція Гевісайда): 1 при **x** ≥ 0; 0 при **x** < 0;

sign(**x**) — повертає знак числа **x**: 0, якщо **x** = 0; 1, якщо **x** > 0; −1, якщо **x** < 0.

Статистичні функції та функції аналізу даних:

corr(A, B) — коефіцієнт кореляції елементів масивів **A** і **B** за Пірсоном. Обчислюється як $\frac{\mathbf{cvar(A,B)}}{\mathbf{stdev(A)\cdot stdev(B)}}$;

cvar(A, B) — коваріація елементів масивів **A** і **B** розмірністю **m×n**,

$$\frac{1}{\mathbf{m\cdot n}}\cdot\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[\mathbf{A_{i,j}}-\mathbf{mean(A)}\right]\cdot\overline{\left[\mathbf{B_{i,j}}-\mathbf{mean(B)}\right]};$$

gmean(A,B,C,...) — повертає середнє геометричне для аргументів **A, B, C, ...** розмірністю **m×n** за виразом $\sqrt[\mathbf{m\cdot n}]{\prod_{i=0}^{m-1}\prod_{j=0}^{n-1}\frac{1}{\mathbf{M_{i,j}}}}$, де **M** — масив, утворений аргументами **A, B, C, ...**;

hist(int, d) — повертає вектор значень частот, з якими величини, що містяться у векторі **d**, потрапляють до інтервалів, представлених межами, заданими у векторі **int**;

histogram(n, d) — повертає матрицю, що має дві колонки: перша містить середини **n** підінтервалів діапазону рівної довжини, а друга є ідентичною обчисленню функції **hist(int, d)**;

hmean(A,B,C,...) — повертає середнє гармонічне для аргументів **A, B, C, ...** за формулою $\left(\frac{1}{\mathbf{m\cdot n}}\cdot\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\frac{1}{\mathbf{M_{i,j}}}\right)^{-1}$;

kurt(A, B, C, ...) — повертає ексцес аргументів **A, B, C, ...** за формулою

$$\left[\frac{\mathbf{m\cdot n\cdot(m\cdot n+1)}}{\mathbf{(m\cdot n-1)\cdot(m\cdot n-2)\cdot(m\cdot n-3)}}\cdot\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left(\frac{\mathbf{M_{i,j}}-\mathbf{mean(M)}}{\mathbf{Stdev(M)}}\right)^4\right]-\frac{\mathbf{3\cdot(m\cdot n-1)^2}}{\mathbf{(m\cdot n-2)\cdot(m\cdot n-3)}},$$

причому масив **M**, утворений на основі аргументів **A, B, C, ...**, повинен мати не менше 4-х елементів, а стандартне відхилення цих елементів $\sigma\neq 0$;

mean(A, B, C, ...) — повертає середнє арифметичне для аргументів **A, B, C, ...**:

$$\frac{1}{\mathbf{m\cdot n}}\cdot\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\mathbf{M_{i,j}};$$

median(A,B,C,...) — повертає медіану для аргументів **A, B, C, ...**;

mode(A, B, C, ...) — повертає значення елемента аргументів **A,B,C,...**, яке зустрічається найбільш часто;

skew(A, B, C, ...) — повертає коефіцієнт асиметрії значень аргументів **A, B, C, ...**, обчислений як

$$\frac{\mathbf{m\cdot n}}{\mathbf{(m\cdot n-1)\cdot(m\cdot n-2)}}\cdot\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left(\frac{\mathbf{M_{i,j}}-\mathbf{mean(M)}}{\mathbf{Stdev(M)}}\right)^3;$$

stderr(u, v) — повертає середньоквадратичну помилку, що відповідає простій лінійній регресії для точок, описаних векторами **u** та **v** (для кожного вектора кількість елементів $\mathbf{n} \geq 3$). При цьому визначається, наскільки близько результати обробки даних розміщені до лінії регресії,

$$\sqrt{\frac{1}{n-2}\cdot\sum_{i}[v_i-\mathbf{intercept}(u,v)+\mathbf{slope}(u,v)\cdot u_i]^2};$$

Stdev(A, B, C, ...) — повертає значення вибіркового стандартного відхилення значень аргументів, як корінь квадратний з дисперсії **Var**(A, B, C, ...);

stdev(A, B, C, ...) — повертає значення стандартного відхилення генеральної сукупності значень аргументів, як корінь квадратний з дисперсії **var**(A, B, C, ...);

Var(A, B, C, ...) — повертає значення вибіркової дисперсії значень аргументів, обчислене як

$$\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(M_{i,j}-\mathbf{mean}(M))^2 \Big/ (m\cdot n-1).$$

Ділення квадратів відхилення на «об'єм вибірки мінус один» дозволяє одержати кращу оцінку істинної дисперсії генеральної сукупності;

var(A, B, C, ...) — повертає значення дисперсії генеральної сукупності, обчислене за формулою

$$\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(M_{i,j}-\mathbf{mean}(M))^2 \Big/ m\cdot n.$$

Таким чином, $\mathbf{var}\cdot\frac{m\cdot n}{m\cdot n-1}=\mathbf{Var}$.

Функції статистичних розподілів і генераторів випадкових чисел:

Позначення наступних функцій здійснені наступним чином: **d...** — функція щільності розподілу ймовірності (ймовірність того, що випадкова величина прийме значення **x**); **p...** — функція кумулятивного розподілу ймовірності (ймовірність того, що випадкова величина прийме значення $\mathbf{X} \leq \mathbf{x}$); **q...** — функція оберненого кумулятивного розподілу ймовірності — квантиль (значення **x**, при якому ймовірність не перевищує задане значення **p**); **r...** — функція, що повертає вектор з **m** випадкових чисел, що мають розподіл «...».

β-розподіл:

$$\frac{\Gamma(s1+s2)}{\Gamma(s1)\cdot\Gamma(s2)}\cdot x^{s1-1}\cdot(1-x)^{s2-1},$$

при параметрах форми $(s1, s2) > 0$ та $0 < x < 1$: **dbeta**(x, s1, s2); **pbeta**(x, s1, s2); **qbeta**(p, s1, s2); **rbeta**(m, s1, s2).

Біноміальний розподіл:

$$\frac{n!}{k!\cdot(n-k)!}\cdot p^k\cdot(1-p)^{n-k},$$

де **n** — кількість незалежних випробувань з імовірністю успіху **p**, $0 \le p \le 1$; **q** — імовірність успіху при однократному випробуванні; **k** — кількість успіхів, $0 \le k \le n$: **dbinom(k, n, p)**; **pbinom(k, n, p)**; **qbinom(p, n, q)**; **rbinom(m, n, p)**.

Розподіл Коші:

$$\left[\pi\cdot s\cdot\left[1+\left(\frac{x-a}{s}\right)^2\right]\right]^{-1},$$

де **a** — параметр розташування; $s > 0$ — параметр масштабу; **dcauchy(x, a, s)**; **pcauchy(x, a, s)**; **qcauchy(p, a, s)**; **rcauchy(m, a, s)**.

Розподіл χ^2:

$$\frac{\exp(-x/2)}{2\cdot\Gamma(d/2)}\cdot\left(\frac{x}{2}\right)^{d/2-1},$$

де **d** — параметр форми (кількість ступенів свободи), $d > 0$; $x > 0$; **dchisq(x, d)**; **pchisq(x, d)**; **qchisq(p, d)**; **rchisq(m, d)**.

Експоненціальний (показниковий) розподіл:

$$r\cdot\exp(-r\cdot x),$$

де **r** — параметр масштабу, $r > 0$; $x > 0$: **dexp(x, r)**; **pexp(x, r)**; **qexp(p, r)**; **rexp(m, r)**.

F-розподіл Фішера:

$$\frac{\sqrt{d1^{d1}\cdot d2^{d2}}\cdot\Gamma\left(\frac{d1+d2}{2}\right)}{\Gamma\left(\frac{d1}{2}\right)\cdot\Gamma\left(\frac{d2}{2}\right)}\cdot\sqrt{\frac{x^{d1-2}}{(d1+d2\cdot x)^{d1+d2}}},$$

де $d1, d2 > 0$ — кількості ступенів свободи; $x > 0$: **dF(x, d1, d2)**; **pF(x, d1, d2)**; **qF(p, d1, d2)**; **rF(m, d1, d2)**.

γ-розподіл:

$$x^{s-1}\cdot\exp(-x)/\Gamma(s),$$

де $s > 0$ — параметр форми; $x \ge 0$: **dgamma(x, s)**; **pgamma(x, s)**; **qgamma(p, s)**; **rgamma(m, s)**.

Геометричний розподіл:

$$p\cdot(1-p)^n,$$

де $0 < p < 1$ — імовірність успіху; **n** — кількість випробувань, $n \ge 0$: **dgeom(n, p)**; **pgeom(n, p)**; **qgeom(p, q)**; **rgeom(m, p)**.

Гіпергеометричний розподіл:

$$C_a^m\cdot\frac{C_b^{n-m}}{C_{a+b}^n} = \text{combin}(a, m)\cdot\frac{\text{combin}(b, n-m)}{\text{combin}(a+b, n)},$$

де **a** + **b** — популяція елементів, **a** з яких мають деяку властивість (видобування такого елемента є успіхом); **n** — обсяг вибірки з популяції без повернення; **m** — кількість успіхів у виборці. Необхідна умова $\max(0, n-b) \le m \le \min(n, a)$: **dhypergeom**(**m**, **a**, **b**, **n**) і **phypergeom**(**m**, **a**, **b**, **n**), для яких $0 \le m \le a$, $0 \le (n-m) \le b$, $0 \le n \le (a+b)$; **qhypergeom**(**p**, **a**, **b**, **n**) і **rhypergeom**(**m**, **a**, **b**, **n**), для яких $0 \le p < 1$, $m > 0$, $(a, b) \ge 0$, $0 \le n \le (a+b)$.

Логнормальний розподіл:

$$\frac{1}{\sqrt{2\cdot\pi}\cdot\sigma\cdot x}\cdot\exp\left[-\frac{1}{2\cdot\sigma^2}\cdot(\ln(x)-\mu)^2\right],$$

де μ — натуральний логарифм від середнього значення; $\sigma > 0$ — параметр форми (натуральний логарифм середньоквадратичного відхилення); $x > 0$: **dlnorm**(**x**, μ, σ); **plnorm**(**x**, μ, σ); **qlnorm**(**p**, μ, σ); **rlnorm**(**m**, μ, σ).

Логістичний розподіл:

$$\exp\left(\frac{a-x}{s}\right)\Big/\left[s\cdot\left(1+\exp\left(\frac{a-x}{s}\right)\right)^2\right],$$

де **a** — параметр розташування; $s > 0$ — параметр масштабу: **dlogis**(**x**, **a**, **s**); **plogis**(**x**, **a**, **s**); **qlogis**(**p**, **a**, **s**); **rlogis**(**m**, **a**, **s**).

Від'ємний біноміальний розподіл:

$$C^k_{n+k-1}\cdot p^n\cdot(1-p)^k = \mathbf{combin}(n+k-1, k)\cdot p^n\cdot(1-p)^k,$$

де $0 < p \le 1$ — імовірність успіху; $x > 0$ — кількість успіхів; $k \ge 0$ — кількість невдач: **dnbinom**(**k**, **n**, **p**); **pnbinom**(**k**, **n**, **p**); **qnbinom**(**p**, **n**, **q**); **rnbinom**(**m**, **n**, **p**).

Нормальний (гаусівський) розподіл:

$$\frac{1}{\sqrt{2\cdot\pi}\cdot\sigma}\cdot\exp\left[-\frac{1}{2\cdot\sigma^2}\cdot(x-\mu)^2\right],$$

де μ — параметр розташування (математичне очікування); $\sigma > 0$ — параметр масштабу (середньоквадратичне відхилення): **dnorm**(**x**, μ, σ); **pnorm**(**x**, μ, σ); **qnorm**(**p**, μ, σ); **rnorm**(**m**, μ, σ); **cnorm**(**x**) — те саме, що й функція **pnorm** при $\mu = 0$ і $\sigma = 1$ (стандартизований нормальний розподіл).

Розподіл Пуассона:

$$\lambda^k\cdot\exp(-\lambda)/k!,$$

де $\lambda > 0$ — параметр розподілу; $k \ge 0$: **dpois**(**k**, λ); **ppois**(**k**, λ); **qpois**(**k**, λ); **rpois**(**m**, λ).

t-розподіл Стьюдента:

$$\frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)\cdot\sqrt{\pi\cdot d}}\cdot\left[\sqrt{\left(1+\frac{x^2}{d}\right)^{d+1}}\right]^{-1},$$

де $d > 0$ — кількість ступенів свободи; **x** – дійсне: **dt**(**x**, **d**); **pt**(**x**, **d**); **qt**(**p**, **d**); **rt**(**m**, **d**).

Рівномірний (прямокутний) розподіл:

$$(b-a)^{-1},$$

де **a** і **b** — відповідно, нижня й верхня границі області значень, причому **a** < **b**, **a** ≤ **x** ≤ **b**: **dunif(x, a, b)**; **punif(x, a, b)**; **qunif(x, a, b)**; **runif(m, a, b)**; **rnd(x)** — повертає рівномірно розподілене випадкове число на інтервалі між 0 і **x**.

Розподіл Вейбулла:

$$s \cdot x^{s-1} \cdot \exp\left(-x^{s}\right),$$

де **s** > 0 — параметр форми; **x** > 0: **dweibull(x, s)**; **pweibull(x, s)**; **qweibull(p, s)**; **rweibull(m, s)**.

Seed(x) — скидає значення початкового числа при генерації ПВЧ в **x**, повертаючи попереднє значення. Причому $1 \le x \le 2147483647$, ціле.

Функції комбінаторного аналізу й теорії чисел:

combin(n, k) — повертає кількість підмножин (комбінацій), розмірністю **k** кожна, що може бути утворена з **n** об'єктів, — еквівалент $\frac{n!}{k!\cdot(n-k)!}$, де $n \ge k$; $n, k \ge 0$;

gcd(A, B, C, ...) — повертає найбільший спільний дільник — найбільше число, на яке без залишку діляться кожен з аргументів **A**, **B**, **C**, ... ;

lcm(A, B, C, ...) — повертає найменше спільне кратне — найменше додатне ціле, яке без залишку ділиться на кожен з аргументів **A**, **B**, **C**, ... ;

mod(x, y) — залишок від ділення **x** на **y** (аргументи мають бути дійсними, $y \ne 0$; результат має той самий знак, що й **x**);

permut(n, k) — повертає кількість можливих способів обрання (перестановки) **k** різних об'єктів з **n** існуючих — еквівалент $n!/(n-k)!$, де $n \ge k$; $n, k \ge 0$.

Функції інтерполяції та екстраполяції:

cspline(vx, vy) — повертає вектор **vs** других похідних для даних, представлених векторами **vx** і **vy**. Побудована за цим вектором сплайнова крива є кубічною у кінцевих точках;

interp(vs,vx,vy,x) — повертає значення сплайн-інтерпольованої функції для векторів **vx** (впорядкованого за зростанням) і **vy** опорних точок та аргументу **x** на основі вектора **vs**. Вектори **vx** і **vy** мають однакову розмірність;

linterp(vx, vy, x) — повертає значення лінійно-інтерпольованої функції для заданих векторів **vx** (впорядкованого за зростанням) і **vy** опорних точок та заданого аргументу **x**. Вектори **vx** і **vy** мають однакову розмірність. При цьому точки на графіку, одержані за даними векторів, з'єднуються відрізками прямих;

lspline(vx, vy) — те саме, що й **cspline**, але сплайнова крива у кінцевих точках є лінійною;

predict(v, m, n) — повертає вектор з **n** лінійно прогнозованих значень на основі **m** послідовних елементів з вектора даних **v**;

pspline(vx, vy) — те саме, що й **cspline**, але сплайнова крива в кінцевих точках є параболічною.

Функції регресії:

expfit(vx, vy, [vg]) — експонентна регресія даних, визначених векторами **vx** і **vy**. Вектор **vg**, якщо використовується, містить прогнозовані значення параметрів **a**, **b** і **c** у степеневому рівнянні $\mathbf{a}\cdot\mathbf{exp}(\mathbf{b}\cdot\mathbf{x})+\mathbf{c}$;

genfit(vx,vy,vg,F) — повертає вектор **K** параметрів функції **F**, що дають мінімальну середньоквадратичну похибку опису функцією $\mathbf{F}(\mathbf{x}, K_1, ..., K_n)$ первинних даних **vx**, **vy**. Вектор **vg** має містити прогнозовані значення **n**-параметрів;

intercept(vx, vy) — повертає значення вільного члену **a** лінійної регресії $\mathbf{a}+\mathbf{b}\cdot\mathbf{x}$;

lgsfit(vx, vy, vg) — регресія логістичною функцією $\mathbf{a}/(1+\mathbf{b}\cdot\mathbf{exp}(-\mathbf{c}\cdot\mathbf{x}))$;

line(vx, vy) — повертає вектор коефіцієнтів лінійної регресії функцією $\mathbf{a}+\mathbf{b}\cdot\mathbf{x}$;

linfit(vx, vy, F) — повертає вектор коефіцієнтів лінійної регресії загального виду. Вектор **F** має містити функції $\mathbf{F}_1(\mathbf{x})$, ..., $\mathbf{F}_n(\mathbf{x})$;

lnfit(vx, vy) — регресія логарифмічною функцією $\mathbf{a}\cdot\mathbf{ln}(\mathbf{x})+\mathbf{b}$;

loess(Mx, My, sp) — повертає вектор коефіцієнтів **vs** для регресії відрізками поліномів (використовується в парі з функцією **interp**). Параметр **sp** вказує розмір локальної області наближуваних даних;

logfit(vx, vy, vg) — регресія логарифмічною функцією $\mathbf{a}\cdot\mathbf{ln}(\mathbf{x}+\mathbf{b})+\mathbf{c}$;

medfit(vx, vy) — повертає вектор коефіцієнтів лінійної регресії функцією $\mathbf{a}+\mathbf{b}\cdot\mathbf{x}$, використовуючи медіанну регресію;

pwrfit(vx, vy, vg) — регресія степеневою функцією $\mathbf{a}\cdot\mathbf{x}^{\mathbf{b}}+\mathbf{c}$;

regress(Mx, vy, n) — повертає вектор коефіцієнтів **vs** для поліноміальної регресії при степені полінома **n** (використовується в парі з функцією **interp**);

sinfit(vx, vy, vg) — регресія синусоїдою $\mathbf{a}\cdot\mathbf{sin}(\mathbf{x}+\mathbf{b})+\mathbf{c}$;

slope(vx, vy) — повертає значення кутового коефіцієнта **b** лінійної регресії $\mathbf{a}+\mathbf{b}\cdot\mathbf{x}$.

Функції статистичного згладжування даних:

ksmooth(vx,vy,b) — повертає **m**-мірний вектор згладжених елементів **vy**, обчислених на основі розподілу Гауса. При цьому **vx**, **vy** — **m**-мірні вектори дійсних чисел; **b** — ширина вікна згладжування;

medsmooth(vy,n) — повертає **m**-мірний вектор згладжених елементів **vy**, обчислених з використанням методу «ковзних медіан». Параметр **n** — ширина вікна згладжування;

supsmooth(vx,vy) — повертає **m**-мірний вектор згладжених елементів **vy**, обчислених на основі адаптивної процедури лінійного згладжування методом найменших квадратів;

expsmooth(B, α) — згладжування даних масиву **B** шляхом застосування експонентного згладжування з ваговим коефіцієнтом α.

Функції дискретного перетворення:

cfft(A) — пряме комплексне перетворення Фур'є для масиву комплексних чисел **A** (повертає масив того ж розміру, що й **A**). Якщо **A** є вектором, $c_j = \frac{1}{\sqrt{n}} \cdot \sum_k A_k \cdot \exp[i \cdot (2 \cdot \pi \cdot j/n) \cdot k]$, де **n** — кількість елементів вектора **A**;

CFFT(A) — те саме, але при нормуванні $1/n$ і від'ємному аргументі функції експоненти;

fft(v) — пряме ШПФ для даних, записаних у векторі **v** у вигляді дійсних чисел з 2^n елементами, де **n** — ціле число (повертається вектор розмірністю $2^{n-1}+1$): $c_j = \frac{1}{\sqrt{n}} \cdot \sum_k v_k \cdot \exp[i \cdot (2 \cdot \pi \cdot j/n) \cdot k]$;

FFT(v) — те саме, але при нормуванні $1/n$ і від'ємному аргументі функції експоненти;

icfft(B) — зворотне комплексне перетворення Фур'є, що відповідає **cfft** (повертає масив з розмірністю аргументу **B**):

$$c_j = \frac{1}{\sqrt{n}} \cdot \sum_k B_k \cdot \exp[-i \cdot (2 \cdot \pi \cdot j/n) \cdot k];$$

iCFFT(B) — те саме, але при нормуванні $1/n$ і додатному аргументі функції експоненти;

ifft(u) — зворотне ШПФ, що відповідає **fft**. Вектор **u** має розмір $2^{n-1}+1$, де **n** — ціле число (повертається вектор розмірністю 2^n):

$$c_j = \frac{1}{\sqrt{n}} \cdot \sum_k u_k \cdot \exp[-i \cdot (2 \cdot \pi \cdot j/n) \cdot k];$$

iFFT(u) — те саме, але при нормуванні $1/n$ і додатному аргументі функції експоненти;

wave(v) — пряме вейвлет-перетворення (вектор **v** має складатися з 2^n дійсних елементів, де **n** — ціле число);

iwave(u) — вектор зворотного вейвлет-перетворення (**u** — вектор частотних даних вейвлет-спектра, розмірністю 2^n).

Функції розв'язку рівнянь:

Given — директива, що відкриває блок розв'язку системи рівнянь або нерівностей;

Find(x, y, ...) — вектор скалярних значень **x**, **y**, ..., що дають розв'язок системи рівнянь у блоці, об'явленому директивою **Given** (кількість повертаних значень дорівнює кількості аргументів);

Isolve(M, v) — повертає вектор невідомих **x**, що дають розв'язок системи лінійних алгебраїчних рівнянь виду **M·x = v**;

Maximize(f, x, y, ...) — повертає вектор значень аргументів, за яких функція **f** досягає максимуму (можливе задання додаткових умов у блоці з **Given**);

Minerr (x1, x2, ...) — вектор значень для **x1**, **x2**, ..., що представляють розв'язок системи рівнянь у блоці, об'явленому директивою **Given**, з мінімальною середньоквадратичною похибкою;

Minimize(f, x, y, ...) — повертає вектор значень аргументів, за яких функція **f** досягає мінімуму (можливе задання додаткових умов у блоці з **Given**);

polyroots(v) — повертає вектор усіх коренів полінома, коефіцієнти якого містяться у **v**;

root(f(var), var, [a, b]) — повертає значення змінної **var**, при якій вираз **f** = 0. **[a, b]** — можливий інтервал пошуку кореня.

Функції помилок (інтеграли ймовірності):

erf(x) — функція помилок (функція Лапласа), $\frac{2}{\sqrt{\pi}} \cdot \int_0^x \exp\left(-t^2\right) dt$;

erfc(x) — комплементарна функція помилок, $\frac{2}{\sqrt{\pi}} \cdot \int_x^\infty \exp\left(-t^2\right) dt$,

erfc(x) = 1 – erf(x).

Експоненційна (показникова) та логарифмічні функції:

exp(z) — основа натурального логарифма (число **e**) у степені **z**;

log(z) — десятковий логарифм від аргументу **z**;

log(z, b) — логарифм від аргументу **z** по основі **b**;

ln(z) — натуральний логарифм від аргументу **z**.

Функції повернення типу виразу:

isArray(x) — повертає 1, якщо **x** — масив, і 0 в інших випадках;

isScalar(x) — повертає 1, якщо **x** — скаляр, і 0 в інших випадках;

isString(x) — повертає 1, якщо **x** — тестовий рядок, і 0 в інших випадках.

Функції імпортування й експортування файлів:

APPENDPRN(*file*) — дописує зміст масиву в кінець розмежованого *ASCII*-файлу. Кількість стовпців у масиві має відповідати кількості стовпців в існуючому файлі;

GETWAVİNFO(*file*) — повертає вектор характеристик аудіофайлу формату *WAV* (кількість каналів, частота дискретизації, кількість рівнів квантування, середня кількість байтів на секунду);

READRGB(*file*) — повертає масив з трьох підмасивів, що представляють, відповідно, червону, зелену й синю компоненти кольорового зображення *file* (*BMP*, *GIF*, *JPG*, *TGA*, *PBM*, *PGM*, *PPM* або *TIF* формату);

READ_RED(*file*), **READ_GREEN**(*file*), **READ_BLUE**(*file*) — повертає масив, що відповідає червоній, зеленій і синій компоненті зображення *file*;

READ_HLS(*file*) — повертає масив, що представляє дані про колір зображення *file* (відтінок, яскравість та насиченість);

READ_HLS_HUE(*file*), **READ_HLS_LİGHT**(*file*), **READ_HLS_SAT**(*file*) — повертає масив, що представляє дані про відтінок, яскравість та насиченість кольору зображення *file*;

READ_HSV(*file*) — повертає масив, що представляє значення відтінку кольору, яскравості й насиченості зображення *file*;

READ_HSV_HUE(*file*) **READ_HSV_SAT**(*file*) **READ_HSV_VALUE**(*file*) — повертає масив, що представляє значення відтінку, насиченості та інтенсивності кольору для зображення *file*;

READ_İMAGE(*file*) — повертає масив, що представляє зображення *BMP*, *GIF*, *JPG*, *TGA*, *PCX*, *PBM*, *PGM*, *PPM* або *TIF*-файлу градаціями сірого;

READBİN(*file*, *type*, [**t**]) — повертає масив двійкових значень на основі файлу, типу *byte* (8 бітів, беззнакове ціле), *double* (64 бітів з плаваючою комою) та ін. **t** — додаткові параметри;

READBMP(*file*) — повертає масив, що представляє зображення *BMP*-формату *file* градаціями сірого;

READFİLE(*file*, *type*, [**t**]) — повертає масив даних на основі файлу типу *delimited* (значення відокремлені символом табуляції), *fixed* (стовпчики мають фіксовану ширину) або *Excel* (файл-таблиця *MS Excel*). **t** — додаткові параметри;

READPRN *file*) — повертає масив значень з файлу *file.prn*;

READWAV(*file*) — повертає масив, кожен стовпчик якого є окремим каналом, а кожен рядок відповідає моменту часу, що визначається номером відліку і частотою дискретизації аудіосигналу *file*;

WRITERGB(*file*) — записує кольорове зображення до файлу;
WRITE_HSV(*file*) — те саме;
WRITE_HLS(*file*) — те саме;
WRITEBIN(*file, type, 1/0*) — записує масив або скаляр до файлу двійкових даних;
WRITEBMP(*file*) — записує зображення у відтінках сірого до файлу;
WRITEPRN(*file*) — записує дані (масив) до текстового файлу;
WRITEWAV(*file*) — записує дані до звукового *WAV*-файлу.

Тригонометричні й гіперболічні функції:

a...(z) — обернена тригонометрична чи гіперболічна функція «...» від **z**;
...h(z) — гіперболічна функція «...» від аргументу **z**;
cos(z) — косинус;
csc(z) — косеканс;
cot(z) — котангенс;
sec(z) — секанс;
sin(z) — синус;
tan(z) — тангенс.

Функції округлення числа:

ceil(z) — повертає найменше ціле, що перевищує або дорівнює **z**;
Ceil(z, y) — повертає найменше кратне **y** число, що $\geq$ **z** (при **y** $\neq$ 0);
floor(z) — повертає найбільше ціле число, що $\leq$ **z**;
Floor(z, y) — повертає найбільше кратне **y** число, що $\leq$ **z** (при **y** $\neq$ 0);
round(z, [n]) — повертає аргумент **z**, округлений до **n** (опціонально) розрядів десяткового дробу. Якщо **n** опущене, повертає аргумент **z**, округлений до найближчого цілого (вважається, що **n** = 0). При цьому функція є аналогічною **floor(z)**, якщо дробова частина менша за 0.5, і **ceil(z)** в іншому випадку. Якщо **n** < 0, повертає аргумент **z**, округлений до **n** знаків ліворуч від десяткової коми;
Round(z, y) — округлює **z** до найближчого кратного **y** шляхом обчислення **round(z/y)·y**;
trunc(z) — повертає цілу частину числа **z** шляхом відкидання мантиси;
Trunc(z, y) — повертає результат обчислення **trunc(z/y)·y**.

Рядкові функції:

concat(S1,S2, ...) — об'єднує рядкові змінні **S1**, **S2**, ... шляхом додавання (конкатенації) рядка **S2** до кінця рядка **S1** і т. д.;
error(S) — виводить повідомлення **S** про помилку;
num2str(z) — повертає рядок, чий символи відповідають десятковому значенню числа **z**;
search(S, sub, m) — здійснює пошук підрядка **sub** у рядку **S**, починаючи з позиції **m**;

str2num(Sn) — перетворює рядкове представлення числа **Sn** на дійсне число. При цьому число може бути комплексним, мати інженерний запис, формат двійкового тощо;

str2vec(S) — перетворює рядок символів на вектор відповідних *ASCII*-кодів;

strlen(S) — повертає кількість символів у рядку **S**;

substr(S, n, m) — починаючи з позиції **n**, виокремлює з рядка **S** підрядок довжиною **m** символів;

vec2str(v) — конвертує елементи вектора **v** *ASCII*-кодів у символьний рядок. Припустимі значення елементів вектора: 9, 10, 13, 32...255.

Векторні та матричні функції:

• *функції об'єднання масивів:*

augment(A, B, ...) — об'єднує масиви **A**, **B**, ... з однаковою кількістю рядків зліва направо;

stack(A, B, ...) — об'єднує масиви **A**, **B**, ... з однаковою кількістю стовпців зверху донизу;

• *функція розділення масивів:*

submatrix(A,m,n,i, j) — повертає з масиву **A** підмасив, що складається з елементів рядків від **m** до **n** та стовпців від **i** до **j**. Необхідна умова: $m \le n$ та $i \le j$;

• *функції створення масивів:*

diag(v) — повертає діагональну матрицю, елементи головної діагоналі якої — вектор **v**;

identity(n) — повертає одиничну матрицю розмірністю $n \times n$;

matrix(m, n, f) — повертає матрицю, **(i, j)**-й елемент якої містить значення заданої попередньо деякої функції **f(i, j)**, де $i = 0, 1, ..., m$ та $j = 0, 1, ..., n$;

• *функції визначення розмірності масивів:*

cols(A) — повертає кількість стовпців у масиві **A**;

last(v) — повертає індекс останнього елемента вектора **v**;

length(v) — повертає загальну кількість елементів у векторі **v**;

rows(A) — повертає кількість рядків у масиві **A**;

• *функції визначення екстремумів значень елементів масивів:*

max(A, B, ...) — повертає найбільший за значенням елемент серед масивів **A**, **B**, ... Якщо значення елементів є комплексними, максимум повертається окремо для дійсної та уявної частин;

min(A, B, ...) — повертає найменший за значенням елемент серед масивів **A**, **B**, ... Якщо значення елементів є комплексними, мінімум повертається окремо для дійсної та уявної частин;

• *функції сортування масивів:*

csort(A, n) — повертає масив **A**, відсортований шляхом перестановки рядків у порядку зростання значень елементів у стовпці **n**. Якщо масив містить комплексні елементи, уявна частина ігнорується (це стосується й наступних функцій сортування!);

reverse(v) — змінює порядок слідування елементів вектора **v** на протилежний;

rsort(A, n) — виконує сортування масиву **A** шляхом перестановки стовпців у порядку зростання значень елементів у рядку **n**;

sort(v) — виконує сортування елементів вектора **v** у порядку зростання їхніх значень;

• *функції пошуку:*

hlookup(z,A,r) — проводить пошук значення **z** у верхньому рядку масиву **A**. У стовпці, в якому знайдено **z**, обирається елемент в рядку **r**. Якщо знайдено декілька значень — повертається вектор (справедливо для всіх функцій пошуку!);

lookup(z, A, B) — виконує пошук значення **z** у масиві **A**. У разі знайдення — повертає значення відповідного (за рядком і стовпцем) елемента масиву **B**;

match(z, A) — проводить пошук у масиві **A** елементів, що мають значення **z**, і повертає індекс (індекси) їхніх позицій в **A**;

vlookup(z,A,c) — проводить пошук значення **z** у лівому стовпці масиву **A**. У рядку, в якому знайдено **z**, обирається елемент в стовпці **c**;

• *функції визначення числа обумовленості матриць:*

cond1(M) — повертає число обумовленості квадратної матриці **M**, обчислене в нормі L_1 як $L_1(M)\cdot L_1(M^{-1}) \xrightarrow{\text{MathCAD}} \text{norm1}(M)\cdot\text{norm1}(M^{-1})$;

cond2(M) — те саме, в нормі L_2;

conde(M) — те саме, в евклідовій нормі;

condi(M) — те саме, у нескінченній нормі;

• *функції визначення норми матриць:*

norm1(M) — повертає L_1-норму як максимум серед результатів підсумовування абсолютних значень елементів стовпців матриці:

$$L_1(M) = \max_j \sum_{i=1}^{m} \left| M_{i,j} \right|;$$

norm2(M) — повертає L_2-норму як корінь квадратний з найбільшого власного значення матриці (λ):

$$L_2(M) = \sqrt{\max(\lambda)} \xrightarrow{\text{MathCAD}} \sqrt{\max\left(\text{eigenvals}\left(M\cdot\overline{M^T}\right)\right)},$$

де $\overline{M^T}$ — спряжена транспонована (можливий зворотний порядок) матриця **M**;

norme(M) — повертає евклідову L_e-норму як корінь квадратний з суми квадратів усіх елементів: $L_e(M) = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m}(M_{i,j})^2}$;

normi(M) — повертає нескінченну L_∞-норму як максимальне значення серед результатів підсумовування абсолютних значень елементів рядків матриці: $L_\infty(M) = \max_i \sum_{j=1}^{m} |M_{i,j}|$;

• *функції визначення рангу та лінійних властивостей матриць:*

geninv(A) — створення лівої, зворотної до **A** (що має повний ранг) матриці, яка задовольняє рівнянню $L \cdot A = E$, де **E** — одинична матриця $n \times n$; **L** — матриця $n \times m$; **A** — матриця $m \times n$, при $m \geq n$. Тобто $L = (A^T \cdot A)^{-1} \cdot A^T$. Якщо **A** — квадратна, несингулярна матриця, то $L = A^{-1}$;

rank(A) — повертає ранг (кількість лінійно незалежних стовпців) масиву **A**;

rref(A) — повертає масив — східчасту форму масиву **A** зі скороченою кількістю рядків;

tr(M) — повертає слід (суму діагональних елементів) квадратної матриці **M**;

• *функції визначення власних векторів і власних значень матриць:*

eigenvals(M) — повертає вектор λ власних значень квадратної матриці **M**, кожен елемент якого задовольняє рівності $M \cdot V^{<i>} = \lambda_i \cdot V^{<i>}$, де $V^{<i>}$ — **i**-й власний вектор матриці **M**. Підсумовування всіх елементів вектора λ еквівалентне функції обчислення сліду **tr(M)**;

eigenvec(M, z) — повертає нормований власний вектор $V^{<i>}$ матриці **M**, що відповідає її власному значенню $z = \lambda_i$;

eigenvecs(M) — повертає матрицю **V**, стовпцями якої є власні вектори матриці **M** (порядок розміщення власних векторів відповідає порядку власних значень λ_i — функція **eigenvals**);

genvals(M, N) — повертає вектор узагальнених власних значень λ'_i матриці **M**, який відповідає матричному виразові $M \cdot V'^{<i>} = \lambda'_i \cdot N \cdot V'^{<i>}$. **M** і **N** — квадратні матриці дійсних елементів;

genvecs(M, N) — повертає матрицю **V'**, стовпці якої містять нормовані узагальнені власні вектори (порядок розміщення векторів відповідає порядку узагальнених власних значень λ'_i, що повертаються функцією **genvals**);

• *функції розкладання матриць:*

cholesky(M) — реалізує розкладання матриці за Холецьким (метод квадратних коренів), повертаючи нижньо-трикутну матрицю **L** (усі еле-

менти над головною діагоналлю якою є нульовими), що задовольняє рівність $\mathbf{L}\cdot\mathbf{L}^{\mathbf{T}} = \mathbf{M}$, де **M** — дійсна, позитивно визначена квадратна матриця. При обчисленні використовується лише верхньо-трикутна частина **M**;

lu(M) — реалізує розкладання квадратної матриці **M** шляхом повернення масиву, що містить три об'єднані по рядках квадратні матриці **P**, **L** і **U**, однакової розмірності з **M**. Дані матриці задовольняють рівнянню $\mathbf{P}\cdot\mathbf{M} = \mathbf{L}\cdot\mathbf{Q}$, де **L** і **U** — відповідно, нижньо- і верхньо-трикутні матриці. Для виділення **P**, **L** і **U** зі спільного масиву треба використати функцію **submatrix** (це ж стосується функцій **qr** і **svd**);

qr(A) — реалізує розкладання масиву **A** розмірністю $\mathbf{m}\times\mathbf{n}$ шляхом повернення масиву, чиї перші **m** стовпців є квадратною ортонормованою матрицею **Q**, що має однакову з **A** кількість рядків, а **n** наступних стовпців містять верхньо-трикутну матрицю **R**, яка задовольняє рівнянню $\mathbf{A} = \mathbf{Q}\cdot\mathbf{R}$;

svd(A) — реалізує розкладання масиву **A** розмірністю $\mathbf{m}\times\mathbf{n}$ ($\mathbf{m} \geq \mathbf{n}$) по його сингулярних числах. Повертає масив, що складається з розміщених одна над одною матриць **U** і **V**, які задовольняють рівнянню $\mathbf{A} = \mathbf{U}\cdot\mathbf{D}\cdot\mathbf{V}^{\mathbf{T}}$, де матриці **U** — верхня, розмірністю $\mathbf{m}\times\mathbf{n}$; **V** — нижня, розмірністю $\mathbf{n}\times\mathbf{n}$; **D** — діагональна, на діагоналі якої розміщені сингулярні числа масиву **A**, одержані за допомогою функції **svds**: $\mathbf{D} \xrightarrow{\text{MathCAD}} \mathbf{diag(svds(A))}$;

svds(A) — повертає вектор сингулярних чисел (корінь квадратний від власних значень матриці $\mathbf{A}^{\mathbf{T}}\cdot\mathbf{A}$) масиву **A**. Останній може мати розмірність $\mathbf{m}\times\mathbf{n}$, $\mathbf{m} \geq \mathbf{n}$;

• *функції одержання логарифмічно рознесених точок:*

logspace(min, max, npts) — повертає вектор з **npts** логарифмічно рознесених точок, починаючи з **min** і завершуючи **max** (**min** і **max** — додатні дійсні числа);

logpts(minexp, dec, dnpts) — повертає вектор, що охоплює **dec** декад з **dnpts** рівновіддалених точок, починаючи з $10^{\mathbf{minexp}}$, по **dnpts** точок на декаду. Кількість елементів результуючого вектора — **dec**·**dnpts**;

• *функція обчислення кореляції векторів:*

correl(vx, vy) — повертає одновимірний коефіцієнт кореляції векторів **vx** і **vy** — вектор довжиною **length(vx)** + **length(vy)** − 1, кожен елемент якого є результатом векторного добутку вектора **vx** із зсунутою версією **vy**;

Деякі додаткові функції обробки масивів зображення:

addnoise(A, p, n) — повертає масив **A** з доданим шумом. Причому до значення пікселя додається або віднімається значення **n** ($0 \le |n| \le 255$) з імовірністю $p/2$ ($0 \le p \le 1$);

and(A, B) / or(A, B) — повертає результат виконання булевих операцій «і»/«або» над масивами зображень **A** і **B** (останні повинні мати однакову розмірність);

augment3(R, G, B) — призначення аналогічне функції **augment**, але обов'язковою є наявність трьох аргументів;

binarize(A, thresh) — повертає двійкову версію масиву **A**: пікселям, значення яких перевищують поріг **thresh**, присвоюється значення 1, іншим — 0;

binarize_auto(A) — те саме, але значення порогу обирається програмою автоматично;

binarize2(A, t1, t2, x, y) — повертає двійкову версію масиву **A**: пікселям, значення яких не виходять за межі **[t1; t2]**, присвоюється значення **x**, іншим — **y**;

blend(A, B) — повертає суміш співрозмірних масивів **A** і **B** шляхом попіксельного обчислення виразу: $A_{i,j} + B_{i,j} - (A_{i,j} \cdot B_{i,j}/255)$;

canny(A, σ, low, high) — повертає двійкове контурне зображення, одержане в результаті виділення контурів зображення за алгоритмом Кенні. $\sigma > 0$ — середньоквадратичне відхилення (зазвичай, не більше 2); **low** < **high** — гістерезисні пороги (дійсні числа);

center(A) — повертає результат ДПФ масиву **A**, трансформований таким чином, що постійна складова знаходиться в центрі;

centsmooth(A) — повертає масив **A**, згладжений за допомогою центрально зваженого елемента розмірністю 3×3;

clip(A, Min, Max) — повертає масив **A**, значення елементів якого обмежені порогами **Min** і **Max**;

colgrad(A) — повертає постовпцевий градієнт (відмінність між стовпцями) масиву **A**;

dct2d(A) — повертає двовимірне швидке *ДКП* масиву **A**;

distform(A, fg) — трансформація масиву **A** за евклідовою відстанню для відтінку сірого **fg**;

equalize(A) — повертає масив **A** зображення з інтенсивністю пікселів шкали рівнів сірого, врегульованою таким чином, щоб формувати лінійну сукупну гістограму;

extract(A, n) — виокремлює підмасив **n**-ї колірної компоненти (**n** = 1_K, 2_G, 3_B) з трикомпонентного масиву **A**;

funmap(A, f) — застосовує функцію однієї змінної **f** до кожного елемента масиву **A**;

getnoise(A) — повертає відмінність між первинним і медіанно-відфільтрованим за допомогою функції **medfilt** масивом **A**;

hist2d(A, B, n) — повертає двовимірну **n**-стовпцеву колірну гістограму для масивів **A** і **B** однакової розмірності;

horzflip(A) — здійснює дзеркальне відображення масиву **A** відносно його вертикальної осі;

idct2d(A) — повертає зворотне двовимірне швидке ДКП масиву **A**;

imhist(A, n) — повертає **n**-стовпцеву колірну гістограму масиву **A**, для значень між 0 і 255 включно (значення, що виходять за цей діапазон, ігноруються);

imhist2(A, n) — повертає **n**-стовпцеву колірну гістограму масиву **A** для наявного діапазону значень інтенсивностей;

immse(A, B) — повертає середньоквадратичну помилку між масивами зображення **A** і **B**;

imquant(A, n) — повертає квантовану версію масиву **A**, що містить лише **n** рівновіддалених один від одного рівнів сірого між 0 і 255;

imquant2(A, v) — повертає квантовану версію **A**, що містить зазначені у векторі **v** рівні сірого;

imsnr(A, B) — повертає відношення «сигнал/шум» між масивами зображення **A** і **B**;

invert(A) — повертає інвертований масив зображення **A** шляхом поелементного віднімання від 255 значень інтенсивностей усіх пікселів;

invert2(A) — повертає масив **A**, нове значення елементів якого обчислюється відніманням від **max(A)** значення поточного елемента й додаванням до одержаного результату **min(A)**;

levelmap(A, v) — повертає масив **A** з інтенсивністю, зміненою на значення елемента вектора **v** ($0 \le \mathbf{v}_i \le 255$), індекс **i** якого відповідає первинній інтенсивності пікселя. Розмірність **v** не має бути меншою за наявне максимальне значення інтенсивності на всій множині пікселів зображення;

mask(A, B) — повертає масив **A**, кожен елемент якого замінюється на нуль, якщо відповідний елемент масиву-маски **B** дорівнює 0;

medfilt(A) — повертає медіанно-відфільтрований масив **A**;

orthosmooth[5](A) — повертає масив **A**, згладжений за допомогою ортогонально зваженого елемента розмірністю 3×3 (або 5×5 для **orthosmooth5**);

putregion(A, B, r, c) — вбудовує масив **B** до масиву **A**, починаючи з рядка **r** і стовпця **c**. Необхідні умови:

$$r + rows(B) - 1 < rows(A),$$
$$c + cols(B) - 1 < cols(A);$$

quantfilt(A, W, quant) — повертає квантильно-відфільтрований масив **A**, використовуючи при цьому суміжну матрицю **W** довільної розмірності та квантиль $0 \le quant \le 1$;

relerror(A, B) — повертає значення відносної похибки між масивами зображення **A** і **B**;

replace(A, B, n) — повертає трикомпонентний масив **A** із заміненим на матрицю **B** підмасивом **n**-ї колірної компоненти. Матриця **B** повинна мати однакову з **A** кількість рядків і рівно третину від кількості стовпців **A**;

rotate(A, α) — обертає масив **A** на **α** градусів проти ходу годинникової стрілки;

rotate90[180,270](A) — обертає масив **A** на 90, 180 або 270° проти ходу годинникової стрілки;

rowgrad(A) — повертає відрядковий градієнт (відмінність між рядками) масиву **A**;

scale(A, Min, Max) — повертає масив **A**, елементи якого пронормовані в діапазоні **[Min; Max]**;

shape_features(A) — повертає матрицю моментів та особливостей відображення кожного окремого пікселя зображення **A**;

threshold(A, thresh) — повертає масив **A**, з присвоєними елементам, значення яких є меншими за поріг **thresh**, нульових значень. Якщо **thresh** є від'ємним числом, нуль присвоюється тим елементам, значення яких перевищують значення **|thresh|**;

translate(A, r, c, pad) — повертає масив однієї розмірності з **A**, причому оригінальні елементи **A** зсунуті на **r** рядків і **c** стовпців, незаповнені елементи нового масиву що при цьому з'явилися, набувають інтенсивності $0 \le pad \le 255$. Знак **r** і **c** визначає напрямок зсуву («плюс» — вниз або праворуч, «мінус» — вгору або ліворуч);

unismooth[5](A) — повертає масив **A**, згладжений за допомогою рівномірно зваженого елемента розмірністю 3×3 (або 5×5 для **unismooth5**);

vertflip(A) — здійснює дзеркальне відображення масиву **A** відносно його горизонтальної осі;

warp(A, T) — застосовує білінійну деформацію представленого масивом **A** зображення, використовуючи точки прив'язки, що містяться в масиві **T**. Останній має розмірність 8×2 і містить, таким чином, 4 пари точок:

zoom(A, hsc, vsc) — повертає масив **A**, масштабований з коефіцієнтами **hsc** і **vsc** (відповідно, по горизонталі й вертикалі). Останні мають бути додатними, ненульовими дійсними числами;

Деякі додаткові функції обробки масивів сигналів

ccepstrum(A) — повертає масив, що є результатом обчислення комплексного кепстра багатоканального сигналу **A**;

cepstrum(v) — повертає кепстр вектора **v**;

chirpz(A, lo, hi, d) — повертає частотний спектр сигналу **A** між **lo** та **hi** з інтервалом частот **d**, причому $0 \le$ **lo** $<$ **hi** ≤ 0.5, а (**hi** – **lo**)/**d** ≥ 2;

costr(A) — повертає результат косинусного перетворення масиву **A**, що містить парну кількість елементів;

icostr(A) — повертає результат зворотного косинусного перетворення масиву **A**;

sintr(A) — повертає результат синусного перетворення масиву **A**, що містить парну кількість елементів;

isintr(A) — повертає результат зворотного синусного перетворення масиву **A**;

phase(A) — повертає масив фаз, обчислених на основі комплексних елементів масиву **A**;

mag(A) — повертає масив амплітуд на основі комплексних елементів масиву **A**;

makeri(Mag, Phase) — повертає масив комплексних чисел на основі амплітудних і фазових елементів, що містяться, відповідно, у співрозмірних масивах **Mag** і **Phase**;

snr(vx, vy, n, r, [w]) — обчислює відношення «сигнал/шум» для векторів сигналу **vx** і **vy**. При цьому вектори діляться на 1 < **n** < **length(vx)** перекривних сегменти з коефіцієнтом перекриття $0 \le$ **r** < 1. Кожен сегмент оброблюється вікном зі смугою **w**;

whiten(n) — повертає **n**-розмірний вектор, елементи якого являють собою рівномірно розподілений білий шум;

convol(A, B) — повертає результат обчислення згортки масивів **A** і **B**, що повинні містити як мінімум два елементи;

deconvol(B, A) — повертає результат обчислення зворотної згортки — знаходження оригіналу **B** на основі **A**.

Додаток Ж.
КОНСТАНТИ *MATHCAD*

Нижче подано найменування математичних і системних констант *Mathcad*, правила їх введення та призначення. Представленим константам можна присвоїти й інші значення: або за допомогою оператора присвоєння (:=) безпосередньо в робочому документі, або через діалогове вікно *Worksheet Options* з меню *Tools*. У дужках подано встановлені значення за умовчанням.

Константа	*Введення*	*Призначення*
π	[Ctrl][Shift]p або p[Ctrl]g	Число π. У числових розрахунках *Mathcad* використовує значення π з урахуванням 16 значущих цифр (3,1415926535897931). У символьних обчисленнях π зберігає своє точне значення
e	e	Основа натурального логарифма (2,7182818 284590451). У символьних обчисленнях e зберігає своє точне значення
∞	[Ctrl][Shift]z	Системна нескінченність (10^{307})
%	%	Відсоток (0,01)
1j (1i)	1j (1i)	Комплексна одиниця ($\sqrt{-1}$)
NaN		Не числове значення
ORIGIN		Нижня границя індексації масивів (0)
TOL		Похибка числових методів різноманітних алгоритмів апроксимації, що визначає умови припинення ітерацій числовим алгоритмом (0,001)
CTOL		Похибка збіжності в блоках розв'язку рівнянь, що обмежує нев'язку, задаючи точність виконання рівнянь (0,001)
Seed		Початкове число при генерації ПВЧ (1)
PRNPRECISION		Кількість десяткових знаків, використовуваних під час запису файлів оператором **WRITEPRN** (4)
PRNCOLWIDTH		Ширина стовпця (кількість символів), що використовується при записі файлів оператором **WRITEPRN** (8)
CWD		Текстова змінна, що зберігає адресу поточного (робочого) документа на диску
FRAME		Змінна лічильника кадрів при роботі з анімаційними рисунками (0)

Додаток 3.
ОПЕРАТОРИ *MATHCAD*

Програмний оператор	*Шаблон*	*Призначення*
Add Line	▪ ▪	Виконує функцію розширення програмного модуля. Розширення фіксується подовженням жирної вертикальної риски програмних модулів за деревовидним принципом. Завдяки цьому можна створювати як завгодно великі програми.
←	▪ ← ▪	Оператор внутрішнього (в тілі програмного модуля) локального привоєння.
if	▪ **if** ▪	Оператор створення умовних виразів: *"вираз"* **if** *"умова"*. Якщо *"умова"* виконується, то повертається значення *"вираз"*. У парі з даним оператором зазвичай використовуються оператори **break** та **otherwise**.
for	**for** ▪ ∈ ▪ ▪	Оператор організації циклів із заданою кількістю повторень: **for** *"змінна"* ∈ **Nmin .. Nmax** *"вираз"* Якщо *"змінна"* в результаті свого інкремента чи декремента з певним кроком належить інтервалу [**Nmin**; **Nmax**], то *"вираз"* виконуватиметься. Сама ж *"змінна"* може бути використана у подальших виразах програмного модуля.
while	**while** ▪ ▪	Оператор організації циклів виконання *"виразу"*, діючих доти, доки виконується задана *"умова"*: **while** *"умова"* *"вираз"*
otherwise	▪ **otherwise**	Оператор альтернативної умови. Зазвичай використовується спільно з оператором **if**. Наприклад, програмний модуль $f(x) := \begin{cases} 23 \text{ if } x < 0.3 \\ 1990 \text{ otherwise} \end{cases}$ повертатиме 23, якщо змінна $x < 0.3$, а в решті випадків — значення 1990.

Програмний оператор	*Шаблон*	*Призначення*
break	**break**	Оператор виклику зупинки роботи програми (циклу). Зазвичай використовується спільно з оператором умовного виразу **if** та операторами циклів **for** чи **while**, забезпечуючи перехід на кінець тіла циклу.
continue	**continue**	Оператор продовження. Використовується для продовження роботи після переривання програми. Використовується в парі з операторами задання циклів **for** чи **while**, забезпечуючи після переривання перехід на початок циклу.
return	**return** ▪	Оператор-функція повернення. Перериває виконання програми й повертає значення свого операнда-аргументу. Наприклад, програмний модуль $f(x) := \begin{cases} \textbf{return } 8 \textbf{ if } x = 1 \\ 1980 \end{cases}$ повертатиме 8 при **x** = 1, а в решті випадків — значення 1980.
on error	▪ **on error** ▪	Оператор обробки помилок. Дозволяє створювати конструкції обробників помилок: *"вираз 1"* **on error** *"вираз 2"*. Якщо при виконанні *"виразу 2"* виникне помилка, то виконуватиметься *"вираз 1"*.

Додаток І.
ТАБЛИЦЯ *ASCII*-КОДІВ

DEC	*BIN*	*Симв.*	*Опис*	*DEC*	*BIN*	*Симв.*	*Опис*
0	00000000	NuL	Не визначений	46	00101110	.	Крапка
1	00000001	☺	Початок заголовку	47	00101111	/	Ліва коса риска
2	00000010	☻	Початок тексту	48	00110000	0	Цифра нуль
3	00000011	♥	Кінець тексту	49	00110001	1	Цифра один
4	00000100	♦	Кінець передачі	50	00110010	2	Цифра два
5	00000101	♣	Запит	51	00110011	3	Цифра три
6	00000110	♠	Підтвердження прийому	52	00110100	4	Цифра чотири
7	00000111	•	Звуковий сигнал	53	00110101	5	Цифра п'ять
8	00001000	◘	Повернення, забій	54	00110110	6	Цифра шість
9	00001001	○	Горизонт. табуляція	55	00110111	7	Цифра сім
10	00001010	◙	Переведення рядка	56	00111000	8	Цифра вісім
11	00001011	♂	Вертик. табуляція	57	00111001	9	Цифра дев'ять
12	00001100	♀	Переведення сторінки	58	00111010	:	Двокрапка
13	00001101	♪	Повернення каретки	59	00111011	;	Крапка з комою
14	00001110	♫	Верхній регістр	60	00111100	<	Знак менше
15	00001111	☼	Нижній регістр	61	00111101	=	Знак рівності
16	00010000	►	Відключ. від лінії	62	00111110	>	Знак більше
17	00010001	◄	Керування 1	63	00111111	?	Знак питання
18	00010010	↕	Керування 2	64	01000000	@	Комерційне "ет"
19	00010011	‼	Керування 3	65	01000001	A	Велика лат. *A*
20	00010100	¶	Керування 4	66	01000010	B	Велика лат. *B*
21	00010101	§	Нема підтвердження	67	01000011	C	Велика лат. *C*
22	00010110	▬	Синхронізація	68	01000100	D	Велика лат. *D*
23	00010111	↨	Кінець перед. блоку	69	01000101	E	Велика лат. *E*
24	00011000	↑	Відміна	70	01000110	F	Велика лат. *F*
25	00011001	↓	Кінець носія	71	01000111	G	Велика лат. *G*
26	00011010	→	Заміна	72	01001000	H	Велика лат. *H*
27	00011011	←	Переривання	73	01001001	I	Велика лат. *I*
28	00011100	∟	Роздільник файлів	74	01001010	J	Велика лат. *J*
29	00011101	↔	Роздільник груп	75	01001011	K	Велика лат. *K*
30	00011110	▲	Роздільник записів	76	01001100	L	Велика лат. *L*
31	00011111	▼	Роздільник елементів	77	01001101	M	Велика лат. *M*
32	00100000		Пробіл	78	01001110	N	Велика лат. *N*
33	00100001	!	Знак оклику	79	01001111	O	Велика лат. *O*
34	00100010	"	Подвійні лапки	80	01010000	P	Велика лат. *P*
35	00100011	#	Знак числа	81	01010001	Q	Велика лат. *Q*
36	00100100	$	Знак долара	82	01010010	R	Велика лат. *R*
37	00100101	%	Знак проценту	83	01010011	S	Велика лат. *S*
38	00100110	&	Амперсанд	84	01010100	T	Велика лат. *T*
39	00100111	'	Апостроф	85	01010101	U	Велика лат. *U*
40	00101000	(	Ліва кругла дужка	86	01010110	V	Велика лат. *V*
41	00101001	)	Права кругла дужка	87	01010111	W	Велика лат. *W*
42	00101010	*	Зірочка	88	01011000	X	Велика лат. *X*
43	00101011	+	Знак плюс	89	01011001	Y	Велика лат. *Y*
44	00101100	,	Кома	90	01011010	Z	Велика лат. *Z*
45	00101101	-	Знак мінус	91	01011011	[	Ліва квадратна дужка

DEC	BIN	Симв.	Опис	DEC	BIN	Симв.	Опис
92	01011100	\	Зворотна коса риска	144	10010000	Ђ	Мала кирил. *дє*
93	01011101	]	Права квадратна дужка	145	10010001	‘	Ліва одиночна лапка
94	01011110	^	Знак вставки	146	10010010	’	Права одиночна лапка
95	01011111	_	Підкреслення	147	10010011	“	Ліві лапки
96	01100000	`	Тупий наголос	148	10010100	”	Праві лапки
97	01100001	a	Мала лат. *a*	149	10010101	•	Очікув. повідомлення
98	01100010	b	Мала лат. *b*	150	10010110	–	Коротке тире
99	01100011	c	Мала лат. *c*	151	10010111	—	Довге тире
100	01100100	d	Мала лат. *d*	152	10011000		Не визначений
101	01100101	e	Мала лат. *e*	153	10011001	™	Торгівельна марка
102	01100110	f	Мала лат. *f*	154	10011010	љ	Мала кирил. *ле*
103	01100111	g	Мала лат. *g*	155	10011011	›	Права куткова лапка
104	01101000	h	Мала лат. *h*	156	10011100	њ	Мала кирил. *не*
105	01101001	i	Мала лат. *i*	157	10011101	ќ	Мала кирил. *ке*
106	01101010	j	Мала лат. *j*	158	10011110	ћ	Мала кирил. *ше*
107	01101011	k	Мала лат. *k*	159	10011111	џ	Мала кирил. *же*
108	01101100	l	Мала лат. *l*	160	10100000		Нерозривний пробіл
109	01101101	m	Мала лат. *m*	161	10100001	Ў	Вел. кирил. *У кратке*
110	01101110	n	Мала лат. *n*	162	10100010	ў	Мала кирил. *у кратке*
111	01101111	o	Мала лат. *o*	163	10100011	Ј	Велика кирил. *Йе*
112	01110000	p	Мала лат. *p*	164	10100100	¤	Валюта
113	01110001	q	Мала лат. *q*	165	10100101	Ґ	Велика кирил. *Ге*
114	01110010	r	Мала лат. *r*	166	10100110	¦	Розірвана риска
115	01110011	s	Мала лат. *s*	167	10100111	§	Знак параграфу
116	01110100	t	Мала лат. *t*	168	10101000	Ё	Велика кирил. *Ё*
117	01110101	u	Мала лат. *u*	169	10101001	©	Авторське право
118	01110110	v	Мала лат. *v*	170	10101010	Є	Велика кирил. *Є*
119	01110111	w	Мала лат. *w*	171	10101011	«	Ліві куткові лапки
120	01111000	x	Мала лат. *x*	172	10101100	¬	Куткове тире
121	01111001	y	Мала лат. *y*	173	10101101	-	М'який перенос
122	01111010	z	Мала лат. *z*	174	10101110	®	Зареєстр. торг. знак
123	01111011	{	Ліва фігурна дужка	175	10101111	Ї	Велика кирил. *І*
124	01111100	|	Вертикальна риска	176	10110000	°	Градус
125	01111101	}	Права фігурна дужка	177	10110001	±	"Плюс або мінус"
126	01111110	~	Тильда	178	10110010	І	Велика кирил. *І*
127	01111111	⌂	Видалення	179	10110011	і	Мала кирил. *і*
128	10000000	Ъ	Велика кирил. *Дже*	180	10110100	ґ	Мала кирил. *Ге*
129	10000001	Ґ	Велика кирил. *Ге*	181	10110101	µ	Знак мікро (мала мю)
130	10000010	‚	Нижній апостроф	182	10110110	¶	Абзац
131	10000011	ґ	Мала кирил. *ге*	183	10110111	·	Серединна точка
132	10000100	„	Подвійні нижні лапки	184	10111000	ё	Мала кирил. *ё*
133	10000101	…	Горизонт. три крапки	185	10111001	№	Номер
134	10000110	†	Початок виділ. даних	186	10111010	є	Мала кирил. *є*
135	10000111	‡	Кінець виділ. даних	187	10111011	»	Праві куткові лапки
136	10001000	€	Знак євро	188	10111100	ј	Мала кирил. *йе*
137	10001001	‰	Знак проміле	189	10111101	Ѕ	Велика кирил. *Зе*
138	10001010	Љ	Велика кирил. *Ле*	190	10111110	ѕ	Мала кирил. *зе*
139	10001011	‹	Ліва куткова лапка	191	10111111	і	Мала кирил. *і*
140	10001100	Њ	Велика кирил. *Не*	192	11000000	А	Велика кирил. *А*
141	10001101	Ќ	Велика кирил. *Ке*	193	11000001	Б	Велика кирил. *Б*
142	10001110	Ћ	Велика кирил. *Ше*	194	11000010	В	Велика кирил. *В*
143	10001111	Џ	Велика кирил. *Же*	195	11000011	Г	Велика кирил. *Г*

DEC	BIN	Симв.	Опис	DEC	BIN	Симв.	Опис
196	11000100	Д	Велика кирил. *Д*	226	11100010	в	Мала кирил. *в*
197	11000101	Е	Велика кирил. *Е*	227	11100011	г	Мала кирил. *г*
198	11000110	Ж	Велика кирил. *Ж*	228	11100100	д	Мала кирил. *д*
199	11000111	З	Велика кирил. *З*	229	11100101	е	Мала кирил. *е*
200	11001000	И	Велика кирил. *И*	230	11100110	ж	Мала кирил. *ж*
201	11001001	Й	Велика кирил. *Й*	231	11100111	з	Мала кирил. *з*
202	11001010	К	Велика кирил. *К*	232	11101000	и	Мала кирил. *и*
203	11001011	Л	Велика кирил. *Л*	233	11101001	й	Мала кирил. *й*
204	11001100	М	Велика кирил. *М*	234	11101010	к	Мала кирил. *к*
205	11001101	Н	Велика кирил. *Н*	235	11101011	л	Мала кирил. *л*
206	11001110	О	Велика кирил. *О*	236	11101100	м	Мала кирил. *м*
207	11001111	П	Велика кирил. *П*	237	11101101	н	Мала кирил. *н*
208	11010000	Р	Велика кирил. *Р*	238	11101110	о	Мала кирил. *о*
209	11010001	С	Велика кирил. *С*	239	11101111	п	Мала кирил. *п*
210	11010010	Т	Велика кирил. *Т*	240	11110000	р	Мала кирил. *р*
211	11010011	У	Велика кирил. *У*	241	11110001	с	Мала кирил. *с*
212	11010100	Ф	Велика кирил. *Ф*	242	11110010	т	Мала кирил. *т*
213	11010101	Х	Велика кирил. *Х*	243	11110011	у	Мала кирил. *у*
214	11010110	Ц	Велика кирил. *Ц*	244	11110100	ф	Мала кирил. *ф*
215	11010111	Ч	Велика кирил. *Ч*	245	11110101	х	Мала кирил. *х*
216	11011000	Ш	Велика кирил. *Ш*	246	11110110	ц	Мала кирил. *ц*
217	11011001	Щ	Велика кирил. *Щ*	247	11110111	ч	Мала кирил. *ч*
218	11011010	Ъ	Велика кирил. *Ъ*	248	11111000	ш	Мала кирил. *ш*
219	11011011	Ы	Велика кирил. *Ы*	249	11111001	щ	Мала кирил. *щ*
220	11011100	Ь	Велика кирил. *Ь*	250	11111010	ъ	Мала кирил. *ъ*
221	11011101	Э	Велика кирил. *Э*	251	11111011	ы	Мала кирил. *ы*
222	11011110	Ю	Велика кирил. *Ю*	252	11111100	ь	Мала кирил. *ь*
223	11011111	Я	Велика кирил. *Я*	253	11111101	э	Мала кирил. *э*
224	11100000	а	Мала кирил. *а*	254	11111110	ю	Мала кирил. *ю*
225	11100001	б	Мала кирил. *б*	255	11111111	я	Мала кирил. *я*

СПИСОК ЛІТЕРАТУРИ

1. Артёхин Б. В. *Стеганография* // Журнал «Защита информации. Конфидент». — 1996. — № 4. — С. 47–50.
2. D. Kahn, *The Codebreakers: The Story of Secret Writing*. Macmillan Publishing Company, New York, USA, 1996. — 1200 p.
3. Хорошко В. О., Азаров О. Д., Шелест М. Є., Яремчук Ю. Є. *Основи комп'ютерної стеганографії* : Навч. посіб. для студентів і аспірантів. — Вінниця : ВДТУ, 2003. — 143 с.
4. Барсуков В. С., Романцов А. П. *Компьютерная стеганография: вчера, сегодня, завтра. Технологии информационной безопасности XXI века*. — матеріали *Internet*-ресурсу «Специальная техника» (http://st.ess.ru/).
5. Грибунин В. Г., Оков И. Н., Туринцев И. В. *Цифровая стеганография*. — М. : «Солон-Пресс», 2009. — 272 с.
6. Auguste Kerckhoffs, *La Cryptographie Militaire. Journal des sciences militaires*. pp: 5–83, Jan. 1883, pp: 161–191, Feb. 1883.
7. Gustavus J. Simmons, *The Prisoner's Problem And The Subliminal Channel*, Advances in Cryptology : Proc. Workshop on Communications Security (Crypto'83, David Chaum, ed.), Plenum Press, 1984, pp: 51–67.
8. Gustavus J. Simmons, *The History of Subliminal Channels*. // IEEE Journal on Selected Areas of Communications. 1998. Vol. 16, № 4. pp: 452–461.
9. Jessica Fridrich, R. Du, M. Long, *Steganalysis of LSB Encoding in Color Images*, Proceedings of ICME 2000, New York City, July 31 – August 2, New York, USA.
10. Birgit Pfitzmann, *Information Hiding Terminology*. In: Information Hiding, Springer Lecture Notes in Computer Science. Vol. 1174, 1996, pp: 347–350.
11. Bruce Schneier, *Applied Cryptography : Protocols, Algorithms, and Source Code in C*, 2nd ed. New York // John Wiley and Sons, 2007. — 784 p.
12. Scott Craver, *On Public-Key Steganography in the Presence of an Active Warden* // Technical report RC 20931, IBM, 1997. 13 p.
13. Ross J. Anderson, *Stretching the Limits of Steganography*. In: Information Hiding, Springer Lecture Notes in Computer Science. V. 1174, 1996, pp: 39–48.
14. Walter Bender, Daniel Gruhl, Norishige Morimoto, Anthony Lu. *Techniques for Data Hiding*. IBM Systems Journal, 35(3&4), pp: 313–336, 1996.
15. Christian Cachin, *An Information-Theoretic Model for Steganography*. In: Information Hiding — 2nd International Workshop, Springer as Lecture Notes in Computing Science. V. 1525, April 1998, pp: 306–318.
16. P. Bassia, Ioannis Pitas, *Robust Audio Watermarking in the Time Domain* // Department of Informatics, University of Thessalonica. In: Signal Processing Conference (EUSIPCO'1998), 9th European. Vol. 1, 8–11 Sept. 1998, pp. 25–28.
17. Richard Popa, *An Analysis of Steganographic Techniques*. The Polytechnic University of Timisoara, Faculty of Automatics and Computers, Department of Computer Science and Software Engineering. 1998. 59 p.

18. Neil F. Johnson, Sushil Jajodia, *Steganalysis of İmages Created using Current Steganography Software*, Workshop on İnformation Hiding Proceedings, Portland Oregon, Springer as Lecture Notes in Computer Science, vol. 1525, 1998.

19. Husrev T. Sencar, Mahalingam Ramkumar, Ali N. Akansu, *Data Hiding Fundamentals And Applications. Content Security İn Digital Multimedia.* ELSEVİER science and technology books, 2004. 364 p.

20. Neil F. Johnson, Zoran Duric, Sushil Jajodia, *İnformation Hiding : Steganography and Watermarking. — Attacks and Countermeasures.* Kluwer Academic Publishers. 2001. 160 p.

21. S. Katzenbeisser, Fabien A. P. Petitcolas (Editors), *İnformation Hiding Techniques for Steganography And Digital Watermark*, Artech House Publishers. 1999. 220 p.

22. Генне О. В. *Основные положения стеганографии* // Журнал «Защита информации. Конфидент», № 3, 2000. — С. 20–26.

23. Кустов В. Н., Федчук А. А. *Методы встраивания скрытых сообщений* // Журнал «Защита информации. Конфидент», № 3, 2000. — С. 34.

24. Быков С. Ф. *Алгоритм сжатия JPEG с позиции компьютерной стеганографии* // Журнал «Защита информации. Конфидент», № 3, 2000. — С. 26.

25. Очков В. Ф. *Mathcad 14 для студентов, инженеров и конструкторов.* — СПб. : «BHV», 2007. — 368 с.

26. Кирьянов Д. В. *Mathcad 15 и Mathcad Prime 1.0.* — СПб.: «BHV», 2012. — 432 с.

27. Kefa Rabah, *Steganography: The Art of Hiding Data*. İnformation Technology Journal 3 (3): pp. 245–269. 2004.

28. Sean Gallagher, *Steganography: How al-Qaeda hid Secret Documents in a porn video*. ArsTechnica, 02 May, 2012. (https://arstechnica.com/business/2012/05/steganography-how-al-qaeda-hid-secret-documents-in-a-porn-video/).

29. K. Maney, *Bin Laden's Messages Could Be Hiding İn Plain Sight*. USA Today, 19 Dec. 2001. (http://www.usatoday.com/life/cyber/ccarch/2001/12/19/maney.htm).

30. R. Mohanakrishnan, *Steganography: Snake İn the Grass*. The Hindu Newspaper (İndia), Dec. 6, 2001.

31. Donald L. Schilling, editor, *Meteor Burst Communications: Theory And Practice*. Wiley series in Telecommunications, New York, USA: Wiley, 1993. 480 p.

32. John H. Nugent, Mahesh S. Raisinghani, *The İnformation Technology And Telecommunications Security İmperative: İmportant İssues And Drivers*, Journal of Electronic Commerce Research, Vol. 3, № 1, 2002, pp: 1–14.

33. Закон України *«Про захист інформації в інформаційно-телекомунікаційних системах»*, № 80/94-ВР від 19.04.2014.

34. Анин Б. Ю. *Защита компьютерной информации.* — СПб. : «BHV», 2000. — 384 с.

35. Соколов А. В., Шаньгин В. Ф. *Защита информации в распределённых корпоративных сетях и системах.* — М. : «ДМК», 2002. — 656 с.

36. Хорошко В. О., Огаркова І. М., Чирков Д. В., Голего А. Г., Горохова Т. Б. *Термінологічний довідник з питань технічного захисту інформації.* / За ред. проф. Хорошка В. О. — 3-є вид., доп. і перероб. — К. : ТОВ «Поліграф-Консалтинг», 2003. — 286 с.

37. Голубєв В. О., Гавловський В. Д., Цимбалюк В. С. *Інформаційна безпека: проблеми боротьби зі злочинами у сфері використання комп'ютерних технологій.* — Запоріжжя : «Просвіта», 2001. — С. 198–201.

38. Інформаційний ресурс дослідницької групи *«CNews Analytics»* (http://www.cnews.com/).

39. Інформаційний ресурс дослідницького центру *«DataPro Research Corporation»*. (http://www.datapro.com/)

40. Neil F. Johnson, Sushil Jajodia, *Steganalysis: The Investigation of Hidden Information*, IEEE Information Technology Conference, Syracuse, New York, USA, Sept. 1st–3rd. 1998.

41. Svyatoslav Voloshynovskiy, S. Pereira, V. Iquise, T. Pun, *Attack Modelling: Towards a Second Generation Watermarking Benchmark.* // Preprint. University of Geneva, 2001. 58 p.

42. Грибунин В. Г. *Критерии оценки надёжности паролей.* — М. : РУСКАРД. 2003.

43. Офіційний сайт Національного Інституту Стандартів і Технологій (НІСТ) США — http://www.nist.gov/.

44. M. Ramkumar, *Data Hiding in Multimedia.* PhD Thesis. New Jersey Institute of Technology, 1999. 72 p.

45. C. E. Shannon, *A Mathematical Theory of Communication.* Bell System Technical Journal, 27 (1948), pp: 379–423, 623–656.

46. Marvel L. *Image Steganography for Hidden Communication.* PhD Thesis. University of Delaware, 1999. 115 p.

47. J. Cox, M. Miller, A. McKellips, *Watermarking as Communications With Side Information* // Proceedings of the IEEE. 1999. Vol. 87. № 7. pp: 1127–1141.

48. Барсуков В. С. *Стеганографические технологии защиты документов, авторских прав и информации* // Обзор специальной техники. — 2000. — № 2. — С. 31–40.

49. M. Kutter, F. A. P. Petitcolas, *A Fair Benchmark For Image Watermarking Systems.* Electronic Imaging'99. Security and Watermarking of Multimedia Contents. V. 3657, Sans Jose, CA, USA, 25-27 January 1999.

50. Khalid Sayood, *Introduction to Data Compression.* Ch. 7, p. 142. Morgan Kaufmann Publishers, 1996.

51. Ahmet M. Eskicioglu, Paul S. Fisher. *Image Quality Measures and their Performance.* IEEE Transactions on Communication, 43(12): 2959–2965, December 1995.

52. P. R. R. L. Nunes, A. Alcaim, M. R. L. Fragoso da Silva. *Quality Measures of Compressed Images for Classification Purposes.* Technical Report CCR-146, IBM Brasil, Rio Scientific Center, P.O. Box 4624, 20.0001 Rio de Janeiro, Brazil, October 1992.

53. S. Winkler. *A Perceptual Distortion Metric for Digital Color Video*. İn: SPİE Proceedings of Human Vision and Electronic İmaging. Vol. 3644, San Jose, CA, January, 1999.

54. S. Winkler. *A Perceptual Distortion Metric for Digital Color Images*. İn: Proc. İCİP, vol. 3, pp. 399–403, Chicago, İL, October 1998.

55. Christian J. van den Branden Lambrecht, Joyce E. Farrell. *Perceptual Quality Metric for Digitally Coded Color Images*. İn: Proceeding of EUSİPCO, pp. 1175–1178, Trieste, İtaly, September 1996.

56. S. J. P. Westen, R. L. Lagendijk, J. Biemond. *Perceptual İmage Quality Based on a Multiple Channel HVS Model*. İn: Proceeding of İCASP, vol. 4, pp. 2351–2354, 1995.

57. I. J. Cox, J. Kilian, F. T. Leighton, T. Shamoon. *Secure Spread Spectrum Watermarking for Multimedia*. İEEE Trans. on İmage Processing, vol. 6, No. 12, December 1997.

58. Luis von Ahn, Nicholas J. Hopper, *Public-Key Steganography*. İn: Advances in Cryptology: Eurocrypt'2004 (C. Cachin and J. Camenisch, eds.), vol. 3027 of Lecture Notes in Computer Science, pp. 322–339, Springer, 2004.

59. M. Backes, C. Cachin, *Public-Key Steganography with Active Attacks*. İBM Research Zurich Research Laboratory, CH-8803, Ruschlikon, Switzerland, August 26, 2004.

60. Шеннон К. *Работы по теории информации и кибернетики*. / Пер. с англ. — М. : «Иностр. литература», 1963. — 829 с.

61. P. Moulin, J. A. O'Sullivan, *İnformation-Theoretic Analysis of İnformation Hiding*. İEEE Transactions on İnformation Theory, vol. 49, № 3, pp. 563–593, March 2003.

62. J. K. Su, J. J. Eggers, B. Girod, *Analysis of Digital Watermarks Subjected to Optimum Linear Filtering and Additive Noise* // Signal Processing. Special İssue on İnformation Theoretic İssues in Digital Watermarking, vol. 81, № 6, pp. 1141–1175, 2001.

63. F. Petitcolas, R. J. Anderson, M. G. Kuhn, *İnformation Hiding — A Survey* // Proceedings İEEE, Special İssue on İdentification and Protection of Multimedia İnformation, vol. 87, № 7, pp. 1069–1078, 1999.

64. F. Hartung, M. Kutter, *Multimedia Watermarking Techniques* // Proceedings İEEE, Special İssue on İdentification and Protection of Multimedia İnformation, vol. 87, № 7, pp. 1079–1107, 1999.

65. Скляр Б. *Цифровая связь. Теоретические основы и практическое применение* / Бернард Скляр; пер. с англ. под ред. А. В. Назаренко. — [2-е изд., исправл.] — М. : «Вильямс», 2003. — 1104 с.

66. M. D. Swanson, M. Kobayahi, A.H. Tewfik, Multimedia *Data-Embedding and Watermarking Strategies*. // Proceeding of İEEE, vol. 86, № 6, pp. 1064–1087, 1998.

67. T. Basar, G. J. Olsder, *Dynamic Noncooperative Game Theory* // SİAM Classics in Applied Mathematics, Philadelphia, PA: SİAM, 1999.

68. T. M. Cover, J. A. Thomas, *Elements of İnformation Theory*, New York: Wiley, 1991.

69. S. İ. Gelfand, M. S. Pinsker, *Coding For Channel With Random Parameters*. Probl. Contr. İnform. Theory, vol. 9, № 1, pp. 19–31, 1980.

70. Игнатов В. А. *Теория информации и передачи сигналов.* — М. : «Радио и связь», 1991. — 280 с.

71. Abraham D. Wyner, *The Wire-tap Channel.* // Bell System Tech. J. 1975. Vol. 54, № 8, pp. 1355–1387.

72. Abraham D. Wyner, J. Ziv, *The Rate-Distortion Function For Source Coding With Side İnformation At The Decoder*, İEEE Trans. İnform. Theory, vol. İT–22, pp. 1–10, Jan. 1976.

73. Яковлев В. А. *Защита информации на основе кодового зашумления. Ч. 1. Теория кодового зашумления.* / Под ред. В. И. Коржика. — СПб. : «ВАС», 1993. — 245 с.

74. Henry Hastur, *MandelSteg*, 1994. ftp://ftp.zedz.net/pub/security/steganography/mandsteg/mandsteg-README.txt.

75. Шиффман Х. *Ощущение и восприятие.* Изд. 5-е. — СПб. : «Питер», 2003. — 928 с.

76. Максименко С.Д., Соловієнко В.О. *Загальна психологія* : Навч. посібник. — К. : МАУП, 2000. — 256 с.

77. A. Watson, *The Cortex Transform: Rapid Computation of Simulated Neural İmages* // Computer Vision, Graphics, and İmage Processing. 1987. Vol. 39, № 3, pp. 311–327.

78. S. Moller, A. Pfitzmann, İ. Stirand, *Computer Based Steganography: How İt Works And Why Therefore Any Restriction On Cryptography Are Nonsense, At Best* // İnformation Hiding: First İnternational Workshop "İnfoHiding'96", Springer as Lecture Notes in Computing Science. 1996. Vol. 1174, pp. 7–21.

79. T. Aura, *Practical İnvisibility İn Digital Communication* // İnformation Hiding: First İnternational Workshop "İnfoHiding'96", Springer as Lecture Notes in Computing Science. 1996. Vol. 1174, pp. 265–278.

80. A. Westfeld, A. Pfitzmann, *Attacks on Steganographic Systems. Breaking the Steganographic Utilities EzStego, Jsteg, Steganos, and S-Tools — and Some Lessons Learned* // Proceeding of the Workshop on İnformation Hiding. 1999. — 16 p.

81. J. Fridrich, *A New Steganographic Method For Palette-Based İmage* // Proceedings of the İSBT PİSP conference, Savannah, Georgia, Apr. 1998, pp. 285–289.

82. K. Matsui, K. Tanaka, *Video-steganography: How To secret Embed A Signature İn A Picture* // İMA intellectual properly project proceeding, vol. 1, № 1, 1994, pp. 187–205.

83. Martin Kutter, Frédéric Jordan, Frank Bossen, *Digital Signature Of Color İmages Using Amplitude Modulation* // Proc. of the SPİE Storage and Retrieval for İmage and Video Databases V. 1997. Vol. 3022, pp. 518–526.

84. J. Hernandez, F. Perez-Gonzalez, J. Rodriguez, G. Nieto, *Performance Analysis of a 2-D Multipulse Amplitude Modulation Scheme for Data Hiding and Watermarking of Still İmages* // İEEE Journal on Selected Areas in Communications. 1998. Vol. 16, № 5, pp. 510–525.

85. A. N. Akansu, R. A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subbands and Wavelets (Series in Telecommunications)*, 2nd edition; Academic Press Inc., Boston, MA. 2000. — 499 p.

86. Jian Zhao, Eckhard Koch, *Embedding Robust Labels into Images for Copyright Protection* // Proceeding of the International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Techniques, Munich-Vienna, Verlag, Aug. 1995, pp. 242–251.

87. Eckhard Koch, Jian Zhao, *Towards Robust and Hidden Image Copyright Labeling* // IEEE Workshop on Nonlinear Signal and Image Processing, Greece, June 20–22, 1995. Pp. 123–132.

88. Joshua R. Smith, Barrett O. Comiskey, *Modulation and Information Hiding in Image* // Information Hiding: First International Workshop "InfoHiding'96", Springer as Lecture Notes in Computing Science, 1996, vol. 1174, pp. 207–226.

89. Ioannis Pitas, *A Method for Signature Casting on Digital Images* // International Conference on Image Processing, IEEE Press, Sept. 1996, vol. 3, pp. 215–218.

90. M. T. Sandford, T. G. Handel, J. M. Ettinger, *Data Embedding Method* // Proceeding of the SPIE 2615, Integration issues in large commercial media delivery systems, 1996, pp. 226–259.

91. Коростиль Ю. М., Шелест М. Е. *Принципы построения стеганографических систем со структурной технологией.* // Праці VII міжнародної конференції з автоматичного управління "Автоматика-2000", Львів, 11–15 вересня 2000 р., секція 7, ч.1. — Львів: ДНДІІІ. — С. 273–286.

92. Alan V. Oppenheim, Ronald W. Schafer, *Discrete-Time Signal Processing: New International Edition*, Pearson Education Limited; 3rd edition, Pearson, 2013. — 1056 p.

93. Рабинер Л., Гоулд Б. *Теория и применение цифровой обработки сигналов* // Пер. с англ.; Под ред. Ю. И. Александрова. — М. : «Мир», 1978. — 848 с.

94. Пузиренко О. Ю. Представлення і прогнозування ефективності нового протоколу оцінки якості реалізації розроблюваних алгоритмів комп'ютерної стеганографії / О. Ю. Пузиренко, Д. О. Павроцький, Л. П. Дюжаєв // Радіотехніка. Радіоапаратобудування: Зб. наук. пр. — Вип. 34. — К. : НТУУ «КПІ». 2007. — С. 150–156.

95. Кошкина Н. В. *Стеганография — особенности использования программ на основе метода наименьшего значащего бита.* (http://www.delphikingdom.ru/asp/viewitem.asp?catalogid=1018).

96. Конахович Г. Ф. *Використання пакету Mathcad v.12 для стеганографічного приховання секретних повідомлень в аудіофайлах* / Г. Ф. Конахович, О. Ю. Пузиренко // Захист інформації: Зб. наук. пр. — Вип. 12. — К. : НАУ, 2005. — С. 126–138.

97. *Малая математическая энциклопедия.* // Э. Фрид, И. Пастор, И. Рейман, П. Ревес, И. Ружа. — Буданешт : Изд-во Академии наук Венгрии, 1976. — 694 с.

98. Michael Luby, Charles Rackoff, *How to Construct Pseudorandom Permutations from Pseudorandom Functions.* // SIAM Journal on Computing, vol. 17(2) : pp. 373–386, April 1998.

99. Pat. US 7895347 B2. *Compact encoding of arbitrary length binary objects* / James P. Schneider; Red Hat, Inc. – № US 11/881,527; decl. 27.07.2007; publ. 22.02.2011.

100. Vincent Darmstaedter, Jean-François Delaigle, Jean-Jacques Quisquater, Benoît Michel Macq, *Low Cost Spatial Watermarking* // Computers and Graphics. 1998. Vol. 22. № 4. pp. 417–424.

101. Фильчаков П. Ф. *Справочник по высшей математике*. — К. : «Наукова думка», 1974. — 744 с.

102. Gerrit C. Langelaar, Jan C. A. van der Lubbe, Reginald L. Lagendijk. *Robust Labeling Methods for Copy Protection of Images* // The Society for Imaging Science and Technology (IS&T) and the International Society for Optical Engineering (SPIE). Storage and Retrieval for Image and Video Database V, vol. 3022, pp. 298–309, San Jose, California, USA, February 1997.

103. Dave Benham, Nasir Memon, Boon-Lock Yeo, Minerva Yeung, *Fast Watermarking of DCT-based Compressed Images* // Proceedings of the International Conference on Image Science, Systems and Technology (CISST'97). Las Vegas, Nevada, June 30 – July 3, 1997, vol. 1, pp. 243–252.

104. Chiou-Ting Hsu, Ja-Ling Wu, *Hidden Digital Watermarks in Images* // IEEE Transactions on Image Processing, Vol. 8, No. 1, Jan 1999, pp. 58–68.

105. Jessica Fridrich, *Combining Low-Frequency and Spread Spectrum Watermarking* // Proc. of the SPIE Conference on Mathematics of Data/Image Coding, Compression and Encryption. 1998. Vol. 3456. Pp. 2–12.

106. Н. Хастингс, Дж. Пикок, *Справочник по статистическим распределениям* // Пер. с англ. А. К. Звонкина. — М. : «Статистика», 1980. — 95 с.

107. Г. Корн, Т. Корн. *Справочник по математике для научных работников и инженеров. Определения, теоремы, формулы.* // Пер. с англ. под ред. И. Г. Арамановича. Изд. 2-е. — М. : «Наука», 1970. — 720 с.

108. D. Gruhl, A. Lu, W. Bender, *Echo Hiding*. Information Hiding Workshop. Cambridge, UK, (1996).

109. Земцов А. П. *Методы цифровой стеганографии для защиты авторских прав*. — М. : «LAP Lambert Academic Publishing», 2012. — 148 с.

110. Чваркова И. Л., Тихоненко С. Г., Садов В. С. *Повышение пропускной способности и стойкости стеганографических систем*. — М. : «LAP Lambert Academic Publishing», 2013. — 136 с.

111. Чичварин Н. В., Волосатова Т. М. *Стеганография в САПР*. — М. : «LAP Lambert Academic Publishing», 2014. — 204 с.

112. Конахович Г. Ф., Пузыренко А. Ю. *Компьютерная стеганография. Теория и практика*. — К. : «МК-Пресс», 2006. — 288 с.

113. Дубов Д. В. *Кіберпростір як новий вимір геополітичного суперництва* : монографія. — К. : НІСД, 2014. — 328 с.

114. Видання *«Die Welt»: Різке зростання кіберзлочинності у Європі* — матеріали *Internet*-ресурсу (http://www.dw.com/uk/a-38555191).

115. Конахович Г. Ф., Мачалін І. О., Пузиренко О. Ю. *Теорія електричного зв'язку* : [навч. посіб.]. — [2-е вид., випр. і доп.]. — К. : ТОВ «НВП Інтерсервіс», 2013. — 368 с.

116. Пузиренко О. Ю. *Комп'ютерні системи стеганографічної обробки і захисту інформації у цифровому звуковому мовленні* : дис. на здоб. наук. ступеня кандидата техн. наук : 05.13.05 / Пузиренко Олександр Юрійович. — К., 2012. — 162 с.

117. *Защищённые радиосистемы цифровой передачи информации* / [Сердюков П. Н., Бельчиков А. В., Дронов А. Е. и др.]; под ред. П. Н. Сердюкова. — М. : «АСТ», 2006. — 403 с.

118. *Защита информации в телекоммуникационных системах* / [Г. Ф. Конахович, В. П. Климчук, С. М. Паук, В. Г. Потапов]. — К. : «МК-Пресс», 2005. — 288 с.

119. Бардин К. В. *Проблема порогов чувствительности и психофизические методы*. — М. : «Наука», 1976. — 395 с.

120. Конахович Г. Ф., Пузиренко О. Ю. *Використання пакету Mathcad v.12 для стеганографічного приховання секретних повідомлень у графічних файлах* // Захист інформації. — 2005. — № 3(26). — С. 21–34.

121. Сэломон Д. *Сжатие данных, изображения и звука* / пер. с англ. В. В. Чепыжова. — М. : «Техносфера», 2006. — 368 с.

122. Сергеенко В. С., Баринов В. В. *Сжатие данных, речи, звука и изображений в телекоммуникационных системах* : Учеб. пособие. — М. : «РадиоСофт», 2009. — 360 с.

123. Смирнов А. А. *Методы и средства компьютерной стеганографии с применением сложных дискретных сигналов для защиты информации в компьютерных системах и сетях* / А. А. Смирнов. — Кировоград : «КОД», 2012. — 352 с.

124. *A Tutorial on Cepstrum and Linear Prediction Cepstral Coefficients (LPCCs)* (http://practicalcryptography.com/miscellaneous/machine-learning/tutorial-cepstrum-and-lpccs/).

125. Oscar Forth (Goz). *Autocepstrum accelerate framework* (https://stackoverflow.com/questions/14353869/autocepstrum-accelerate-framework).

126. *Information technology. Coding of moving pictures and associated audio for digital storage media at up to 1,5 Mbit/s. Part 3. Audio:* ISO/IEC 11172-3:1993. — [In force since 1993-01-01]. — 1993. — 150 p.

127. *Інформаційні технології. Метод кодування аудіовізуальної інформації MPEG-1. Частина 3. Аудіо. Цифрове кодування звуку. Загальні технічні вимоги (ISO/IEC 11172-3:1993, IDT)*: ДСТУ *ISO/IEC 11172-3:2003*. — [Чинний від 2006-04-01]. — К. : Держспоживстандарт України, 2005. — 153 с. — (Національний стандарт України).

128. *Information technology. Generic coding of moving pictures and associated audio information. Part 3. Audio:* ISO/IEC 13818-3:1998. — [In force since 1998-04-01]. — 1998. — 121 p.

129. *Інформаційні технології. Метод кодування аудіовізуальної інформації MPEG-2. Частина 3. Аудіо. Цифрове кодування звуку. Загальні технічні вимоги (ISO/IEC 13818-3:1998, IDT)*: ДСТУ *ISO/IEC 13818-3:2003*. — [Чинний від 2006-04-01]. — К. : Держспоживстандарт України, 2005. — 125 с. — (Національний стандарт України).

130. *General methods for the subjective assessment of sound quality:* ITU-R Recommendation BS.1284-1 / International Telecommunications Union. — 2003. — 13 p.

131. Попов О. Б. *Цифровая обработка сигналов в трактах звукового вещания* / О. Б. Попов, С. Г. Рихтер. — М. : Горячая линия-Телеком, 2007. — 341 с.

132. Рабинер Л. Р. *Цифровая обработка речевых сигналов* / Л. Р. Рабинер, Р. В. Шафер; пер. с англ. под ред. М. В. Назарова, Ю. Н. Прохорова. — М. : Радио и связь, 1981. — 496 с.

133. Кинтцель Т. *Руководство программиста по работе со звуком* / Тим Кинтцель; пер. с англ. И. Г. Злобина. — М. : ДМК Пресс, 2000. — 432 с.

134. Hacker S. *MP3. The Definitive Guide* / Scot Hacker. — Sebastopol : O'Reilly, 2000. — 400 p.

135. Ковалгин Ю. А. *Цифровое кодирование звуковых сигналов* / Ю. А. Ковалгин, Э. И. Вологдин. — СПб. : Корона-принт, 2004. — 240 с.

136. Pan D. A *Tutorial on MPEG Audio Compression* / Davis Pan // IEEE Multimedia. — 1995. — Vol. 2, № 2. — P. 60–74.

137. *Radio Broadcasting Systems. Digital Audio Broadcasting to mobile, portable and fixed receivers:* ETSI EN 300 401, v. 1.4.1. — [In force since 2006-06-01] — European Telecommunications Standards Institute, 2006. — 197 p.

138. Пат. 89054 UA, МПК H 04 J 4/00. *Спосіб цифрового радіомовлення з передаванням сервісної інформації стеганографічним каналом передавання даних, створеним на основі аудіоінформації радіопрограм, що транслюються, і пристрій для його реалізації* / Бабак В. П., Конахович Г. Ф., Пузиренко О. Ю.; власник Нац. авіац. ун-т. — № a 2007 02490; заявл. 06.03.2007; опубл. 25.12.2009, Бюл. № 24.

139. Пузиренко О. Ю. *Комп'ютерні системи стеганографічної обробки і захисту інформації у цифровому звуковому мовленні.* — Дис. кандидата техн. наук : 05.13.05 / Пузиренко Олександр Юрійович. — К., 2012. — 159 с.

140. Алдошина И. А. *Основы психоакустики* // Звукорежиссёр. — 1999. — № 6, 7, 9. — 2001. — № 2, 3, 8. — 2002. — №1, 3, 5, 8, 9. — 2003. — № 3.

141. ISO 226:2003. *Acoustics – Normal equal-loudness-level contours.* 2nd ed.

142. *The Computer Engineering Handbook.* 2nd ed. / Vojin G. Oklobdzija, editor-in-chief. — CRC Press, 2007. — 1648 p.

143. Jessica Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications* — 1st ed. — New York : Cambridge University Press, 2009. — 437 p.

144. Nissar A., Mir A. H. *Classification of Steganalysis Techniques: A study* / Arooj Nissar, Ajaz H. Mir // Digital Signal Processing. — 2010. — Vol. 20, № 6. — Pp. 1758–1770.

145. Bin L. et al. *A Survey on Image Steganography and Steganalysis* / Li Bin, He Junhui, Huang Jiwu, Shi Yun Qing. — Journal of Information Hiding and Multimedia Signal Processing. — 2011. — Vol. 2, № 2. — Pp. 142–172.

146. Malik H. et al. *Steganalysis of QIM-based data hiding using kernel density estimation* / Hafiz Malik, K. P. Subbalakshmi, Rajarathnam Chandramouli. — Proceedings of the Multimedia and Security Workshop. — 2007. — Pp. 149–160;

147. Malik H. et al. *Nonparametric Steganalysis of QIM Steganography Using Approximate Entropy* / Hafiz Malik, K. P. Subbalakshmi, Rajarathnam Chandramouli. — IEEE Transactions on Information Forensics and Security. — 2011. — Vol. 7, № 2. — Pp. 418–431.

148. Harmsen J. J. *Steganalysis of additive-noise modelable information hiding* / Jeremiah J. Harmsen, William A. Pearlman. — Proc. SPIE, Security and Watermarking of Multimedia Contents V. — Santa Clara, USA, 2003.

149. Sullivan K. *Steganalysis for Markov Cover Data With Applications to Images* / Kenneth Sullivan, Upamanyu Madhow, Shivkumar Chandrasekaran, B. S. Manjunath. — IEEE Transactions On Information Forensics And Security. — 2006. — Vol. 1, № 2. — Pp. 275–287.

150. Аграновский А. В. *Стеганография, цифровые водяные знаки и стеганализ* / Аграновский А. В., Балакин А. В., Грибунин В. Г., Сапожников С. А. — М. : «Вузовская книга», 2009. — 220 с.

151. Швидченко И. В. *Методы стеганоанализа для графических файлов* // Искусственный интеллект. — 2010. — № 4. — С. 697–705.

152. Fridrich J. *Detecting LSB steganography in color, and gray-scale images* / Jessica Fridrich, Miroslav Goljan, Du Rui. — IEEE Multimedia Magazine, Special Issue on Security. — 2001. — Vol. 8, № 4. — Pp. 22–28.

153. Filler T. *Minimizing additive distortion in steganography using syndrome-trellis codes* / Tomáš Filler, Jan Judas, Jessica Fridrich. — IEEE Transactions on Information Forensics and Security. — 2011. — Vol. 6, № 3. — Pp. 920–935.

154. Filler T. *Steganography using Gibbs random fields* / Tomáš Filler, Jessica Fridrich. — Proceedings of the 12th ACM workshop on Multimedia and Security. — New York, USA, 2010. — Pp. 199–212.

155. Dumitrescu S. *On steganalysis of random LSB embedding in continuous-tone images* / Sorina Dumitrescu, Xiaolin Wu, Nasir Memon. — International Conference on Image Processing. — IEEE, 2002. — Vol. 3. — Pp. 641–644.

156. Pevný T. *Using high-dimensional image models to perform highly undetectable steganography* / Tomáš Pevný, Tomáš Filler, Patrick Bas. — Proceedings of Information Hiding, 12th International Workshop, Lecture Notes in Computer Science. — Ed. Böhme R., Safavi-Naini R. — Calgary, 2010. — Pp. 161–177.

157. Bas P. *"Break Our Steganographic System": The Ins and Outs of Organizing BOSS* / Patrick Bas, Tomáš Filler, Tomáš Pevný. — Proceedings of 13th International Conference, Lecture Notes in Computer Science. — Prague, 2011. — Vol. 6958.

158. Holub V. *Universal distortion function for steganography in an arbitrary domain* / Vojtěch Holub, Jessica Fridrich, Tomáš Denemark. — EURASIP Journal on Information Security. — 2014. — Vol. 1.

159. Pevný T. *Steganalysis by Subtractive Pixel Adjacency Matrix* / Tomáš Pevný, Patrick Bas, Jessica Fridrich. — IEEE Transactions on Information Forensics and Security. — 2010. — Vol. 5, № 2. — Pp. 215–224.

160. Гонсалес Р. *Цифровая обработка изображений* / Рафаэл С. Гонсалес, Ричард Е. Вудс. — Изд. 3-е, исправл. и дополн. — М. : «Техносфера», 2012. — 1104 с.

161. Прогонов Д. О. *Структурні методи пасивного стеганалізу цифрових зображень:* дис. кандидата техн. наук : 05.13.21 / Прогонов Дмитро Олександрович. — К., 2016. — 300 с.

162. Fridrich J. *Rich Models for Steganalysis of Digital Images* / Jessica Fridrich, Jan Kodovský. — IEEE Transactions on Information Forensics and Security. — 2012. — Vol. 7, № 3. — Pp. 868–882.

163. Kodovský J. *Steganalysis of JPEG images using rich models* / Jan Kodovský, Jessica Fridrich. — Proc. SPIE 8303, Media Watermarking, Security, and Forensics. — Editors: Memon Nasir D., Alattar Adnan M., Delp Edward J.

164. Goljan M. *Rich Model for Steganalysis of Color Images* / Miroslav Goljan, Jessica Fridrich, Rémi Cogranne. — Proceedings of IEEE Workshop on Information Forensic and Security. — Atlanta, USA, 2014.

165. Guo L. *An Efficient JPEG Steganographic Scheme Using Uniform Embedding* / Linjie Guo, Jiangqun Ni, Yun Qing Shi. — IEEE International Workshop on Information Forensics and Security. — Tenerife, 2012. — Pp. 169–174.

166. Sedighi V. *Content-adaptive pentary steganography using the multivariate generalized Gaussian cover model* / Vahid Sedighi, Jessica Fridrich, Rémi Cogranne. — Proc. of SPIE Media Watermarking, Security, and Forensics. — Editors: Alattar Adnan M., Memon Nasir D., Heitzenrater Chad D. — 2015.

167. Denemark T. *Improving Steganographic Security by Synchronizing the Selection Channel* / Tomáš Denemark, Jessica Fridrich // Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security. — New York, USA, 2015.

168. Westfeld A. *F5 — A steganographic algorithm: High capacity despite better Steganalysis* / Andreas Westfeld. — Proceedings of 4th International Workshop on Information Hiding, Lecture Notes in Computer Science. — Ed. Moskowitz I. S. — Pittsburgh, USA, 2001. — Pp. 289–302.

169. Provos N. *Defending against statistical steganalysis* / Niels Provos. – Proc. of the 10th USENIX Security Symposium. — Berkeley, USA, 2001. — Vol. 10. — Pp. 323–325.

170. Solanski K. *YASS: Yet Another Steganographic Scheme that resists blind steganalysis* / Kaushal Solanski, Anindya Sarkar, B. S. Manjunath. — 9th International Workshop, IH 2007, Lecture Notes in Computer Science. — Saint Malo, France, 2007. — Vol. 4567. — Pp. 16–31.

171. Sallee P. *Model-Based Steganography* / Phil Sallee. — 2nd International Workshop, IWDW, Lecture Notes in Computer Science. — Seoul, Korea, 2004. — Pp. 154–167.

172. Fridrich J. *Feature-Based Steganalysis for JPEG Images and Its Implications for Future Design of Steganographic Schemes* / Jessica Fridrich. — Proceeding of 6th International Workshop, IH 2004. — Toronto, Canada, 2004. — Pp. 67–81.

173. Pevný T. *Merging Markov and DCT features for multiclass JPEG steganalysis* / Tomáš Pevný, Jessica Fridrich. — Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX. — San Jose, USA, 2007. — Ed. Delp E. J., Wong P. W. — Pp. 1–14.

174. Kodovský J. *Calibration revisited* / Jan Kodovský, Jessica Fridrich. — Proceedings of the 11th ACM workshop on Multimedia and security. — New York, USA, 2009. – Pp. 63–74.

175. Дворкович В. П. *Цифровые видеоинформационные системы (теория и практика)* / Дворкович Виктор Павлович, Дворкович Александр Викторович. — М. : «Техносфера», 2012. — 1008 с.

176. Kodovský J. *Steganalysis in high dimensions: fusing classifiers built on random subspaces* / Jan Kodovský, Jessica Fridrich. — SPIE Proceedings of Media Watermarking, Security, and Forensics III. — Ed. Memon N. D., Dittmann J., Alattar A. M., Delp E. J. — San Francisco, USA, 2011.

177. Dey N. *A Novel Approach of Color İmage Hiding using RGB Color planes and DWT* / Nilanjan Dey, Anamitra Bardhan Roy, Sayantan Dey. — İnternational Journal of Computer Applications. — 2011. — Vol. 36, № 5. — Pp. 19–24.

178. Agarwal R. *Digital watermarking in the singular vector domain* / Rashmi Agarwal, M. S. Santhanam. — İnternational Journal of İmage and Graphics. — 2008. — Vol. 8, № 3. — Pp. 351–362.

179. Joseph A. *Robust Watermarking Based on DWT-SVD* / Anumol Joseph, K. Anusudha. — İnternational Journal on Signal&İmage Security. — 2013. — Vol. 1, № 1.

180. Khan M. İ. *Digital Watermarking for İmage Authentication based on Combined DCT, DWT, and SVD Transformation* / Mohammad İbrahim Khan, Md. Maklachur Rahman, Md. İqbal Hasan Sarker. — İnternational Journal of Computer Science İssues. — 2013. — Vol. 10, № 5. — Pp. 223–230.

181. Elahian A. *İmproved robust DWT-watermarking in YCbCr color space* / Atefeh Elahian, Mehdi Khalili, Shahriar Baradaran Shokouhi. — Global Journal of Computer Application and Technology. — 2011. — Vol. 1, № 3. — Pp. 300–304.

182. Gunjal B. L. *Secured color image watermarking technique in DWT-DCT domain* / Baisa L. Gunjal, Suresh N. Mali. — İnternational Journal of Computer Science, Engineering and İnformation Technology. — 2011. — Vol. 1, № 3. — Pp. 36–44.

183. Красильников Н. Н. *Цифровая обработка 2D- и 3D-изображений* / Красильников Николай Николаевич. — СПб. : «БХВ-Петербург», 2011. — 608 с.

184. Murphy K. P. *Machine Learning: A Probabilistic Perspective* / Kevin P. Murphy. — 1st edition. — The MİT Press, 2012. — 1104 p.

185. Большаков А. А. *Методы обработки многомерных данных и временны́х рядов* / Большаков Александр Афанасьевич, Каримов Равиль Нургалиевич — М. : «Горячая линия – Телеком», 2007. — 522 с.

186. Голяндина Н. Э. *Метод "Гусеница"-SSA: анализ временных рядов* / Голяндина Нина Эдуардовна. — СПб. : СПбГУ, 2004. — 76 с.

187. Фисенко В. Т. *Компьютерная обработка и распознавание изображений* / Фисенко Валерий Трофимович, Фисенко Татьяна Юрьевна. — СПб.: СПбГУ ИТМО, 2008. — 192 с.

188. Nance J. *Periods of the discretized Arnold's Cat Mapping and its extension to n-dimensions* / Joe Nance. — Cornell University Electronic Library, 2016. — 11 p.

189. Holub V. *Random Projections of Residuals for Digital İmage Steganalysis* / Vojtěch Holub, Jessica Fridrich. — İEEE Transactions on İnformation Forensics and Security. — 2013. — Vol. 8, № 12. — Pp. 1996–2006.

190. Avcibaş İ. *Steganalysis using image quality metrics* / İsmail Avcibaş, Nasir Memon, Bülent Sankur. — İEEE Transactions on İmage Processing. — 2003. — Vol. 12, № 2. — Pp. 221–229.

191. Kodovský J. *Ensemble Classifiers for Steganalysis of Digital Media* / Jan Kodovsky, Jessica Fridrich, Vojtěch Holub. — İEEE Transactions on İnformation Forensics and Security. — 2011. — Vol. 7, № 2. — Pp. 432–444.

192. Bishop C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)* / Christopher M. Bishop. — Secaucus : Springer, 2006. — 749 p.

193. Breiman L. *Bagging Predictors* / Leo Breiman. — Technical Report No. 421, Department of Statistics, University of California at Berkeley, 1994. — 19 p.

194. Mathews B. W. *Comparison of the predicted and observed secondary structure of T4 phage lysozyme.* — Biochimica et Biophysica Acta (BBA) - Protein Structure. — BIOCHIM BIOPHYS ACTA PROTEIN. — 1975. — Vol. 405, № 2. — Pp. 442–451.

195. Powers D. M. W. *Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation* / David Martin Ward Powers. — Technical Report. — School of İnformatics and Engineering, Flinders University, Adelaide, Australia. — 2007. — 24 p.

196. Huiskes M. J. *The MİR Flickr Retrieval Evaluation* / Mark J. Huiskes, Michael S. Lew. — Proceedings of the 1st ACM İnternational conference on Multimedia information retrieval. — New York, USA, 2008. — Pp. 39–43.

197. Thomee B. et al. *The New Data and New Challenges in Multimedia Research* / Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, Li-Jia Li. — 2015. — Cornell University Library, Preprint Archive. — arXiv:1503.01817.

198. Ker A. D. et al. *The square root law of steganographic capacity* / Andrew D. Ker, Tomáš Pevný, Jan Kodovský, Jessica Fridrich. — Proceedings of the 10th ACM workshop on Multimedia and security. – New York, USA, 2008. — Pp. 107–116.

199. Lyu S. *Natural İmage Statistics* / Siwei Lyu. — İn «Digital İmage Forensics. There is More to a Picture than meets the Eye», Ed. Sencar H. T., Memon N. — London : Springer, 2013. — Pp. 239–256.

200. Goljan M. *CFA-aware features for steganalysis of color images* / Miroslav Goljan, Jessica Fridrich. — Proceedings of SPİE, Electronic İmaging, Media Watermarking, Security, and Forensics XVII. — San Francisco, USA, 2015.

201. Kodovský J. *Modern steganalysis can detect YASS* / Jan Kodovský, Tomáš Pevný, Jessica Fridrich. — Proc. SPİE 7541, Media Forensics and Security II. — San Jose, USA, 2010. — Ed. Memon N. D., Dittmann J., Alattar A. M., Delp E. J. — Pp. 1–11.

202. Яцура П. П. та ін. *Аналіз методів виявлення демаскуючих ознак стеганограм* / Яцура П. П., Прогонов Д. О., Кущ С. М. — Матеріали XII Всеукраїнської науково-практичної конференції «Теоретичні та прикладні проблеми фізики, математики та інформатики». — *К. : ВПІ ВПК «Політехніка», 2014. — С. 229–231.*

203. Голубничий В. О. та ін. *Демаскуючі ознаки вбудованих стегоданих* / Голубничий В. О., Прогонов Д. О., Кущ С. М. — Матеріали XII Всеукраїнської науково-практичної конференції «Теоретичні та прикладні проблеми фізики, математики та інформатики». — К. : ВІЙ ВПК «Політехніка», 2014. — С. 172–174.

204. Голубничий В. О. та ін. *Виявлення демаскуючих ознак стеганограм, сформованих на основі сингулярного розкладу матриць цифрових зображень* / Голубничий В. О., Прогонов Д. О., Кущ С. М. — Збірник матеріалів Міжнародної науково-технічної конференції «Радіотехнічні поля, сигнали, апарати та системи». — К., 2015. — С. 259–262.

205. Голубничий В. О. та ін. *Використання метрик якості цифрових зображень для виявлення стеганограм, створених на основі комплексних методів приховування повідомлень* / Голубничий В. О., Прогонов Д. О., Кущ С. М. — Матеріали XIII Всеукраїнської науково-практичної конференції «Теоретичні та прикладні проблеми фізики, математики та інформатики». — К. : ВІЙ ВПК «Політехніка», 2015. — С. 155–157.

206. Progonov D. O. *Evaluation of the effectiveness of applying the image quality metrics for acquisition the steganograms* / Progonov D. O., Kushch S. M. — Proceeding of the 3rd International scientific conference "Theoretical and applied aspects of cybernetics" (ТААС'2013). — Ed. Terletskyi D. O., Prashchur A. A., Lytvynenko I. O. — Kyiv, 2013. — Pp. 34–42.

207. Li S. Z. *Markov Random Field Modeling in Image Analysis* / Stan Z. Li. — London : Springer, 2009. — 362 p.

208. Mallat S. *A Wavelet Tour of Signal Processing. The Sparse Way* / Stéphane Mallat. — 3rd Edition. — Burlington : Academic Press, 2008. — 805 p.

209. Materka A. *Texture Analysis Methods – A Review* / Andrzej Materka, Michal Strzelecki. – – COST B11 report. — Institute of Electronics, Technical University of Lodz. — Brussel, 1998. — 33 p.

210. Tuceryan M. *Texture Analysis* / Mihran Tuceryan // The Handbook of Pattern Recognition and Computer Vision. — Ed. Chen C. H., Pau L. F., Wang P. S. P. — 2nd Edition — World Scientific Publishing Co., 1998. — Pp. 207–248.

211. Фраленко В. П. *Методы текстурного анализа изображений, обработка данных дистанционного зондирования Земли* / Фраленко Виталий Петрович. — Программные системы: теория и приложения. — № 4(22). — 2014. — С. 19–39.

212. Zucker S. W. *Multiple-level Representations for Texture Discrimination* / Steven W. Zucker, Kamal Kant. — Proceedings of the IEEE Conference on Pattern Recognition and Image Processing. — Dallas, USA, 1981. — Pp. 609–614.

213. Gelfand A. E. *Handbook of Spatial Statistics* / Alan E. Gelfand, Peter J. Diggle, Montserrat Fuentes, Peter Guttorp (eds). — Boca Raton : Chapman & Hall/CRC, 2010. — 619 p.

214. Cressie N. *Statistics for Spatio-Temporal Data* / Noel Cressie, Christopher K. Wikle. — Wiley, 2011. — 624 p.

215. Plotnick R. E. *Lacunarity indices as measures of landscape texture* / Roy E. Plotnick, Robert H. Gardner, Robert V. O'Neil // Landscape Ecology. — 1993. — Vol. 8, № 3. — Pp. 201–211.

216. Peitgen H.-O. *Chaos and Fractals. New Frontiers of Science* / Heinz-Otto Peitgen, Hartmut Jürgens, Dietmar Saupe. — 2nd Edition. — New York : Springer, 2004. — 864 p.

217. Viton P. A. *Notes on Spatial Econometric Models* / Philip A. Viton. — Technical Report. — The Ohio State University, 2010. — 23 p.

218. Moran P. A. P. *Notes on continuous stochastic phenomena* / Patrick Alfred Pierce Moran. // Biometrika. — 1950. — Vol. 37, № 1–2. — Pp. 17–23;

219. Geary R. C. *The Contiguity Ratio and Statistical Mapping* / Robert Charles Geary. // The Incorporated Statistician. — 1954. — Vol. 5, № 3. — Pp. 115–146.

220. Божокин С. В. *Фракталы и мультифракталы* / Божокин Сергей Валентинович, Паршин Дмитрий Алексеевич. — Ижевск : НИЦ "Регулярная и хаотическая динамика", 2001. — 128 с.

221. Kantelhardt J. W. *Fractal and multifractal time series* / Jan W. Kantelhardt. — Encyclopedia of Complexity and Systems Science. — Ed. Meyers Robert A. — New York : Springer, 2009. — Pp. 3754–3779.

222. Kantelhardt J. W. et al. *Multifractal Detrended Fluctuation Analysis of Nonstationary Time Series* / Jan W. Kantelhardt, Stephan A. Zschiegner, Eva Koscielny-Bunde, Armin Bunde, Shlomo Havlin, H. Eugene Stanley. Physica A: Statistical Mechanics and its Applications. — 2002. — Vol. 316, № 1–4. — Pp. 87–114.

223. Cressie N. *Statistics for Spatial Data. Revised Edition* / Noel Cressie. — Wiley-Interscience, 1993. — 928 p.

224. Smith T. E. *Notebook on Spatial Data Analysis* [*Internet*-ресурс]. — 2016. — Режим доступу: https://www.seas.upenn.edu/~ese502/#notebook

225. Das S. *On the Estimation of Parameters of Variograms of Spatial Stationary Isotropic Random Processes* / Sourav Das, Tata Subba Rao, Georgi N. Boshnakov. — Technical report. — Probability and Statistics Group, School of Mathematics, University of Manchester. — 2012. — 22 p.

226. Hurst H. E. *Long-Term Storage Capacity of Reservoirs* / Harold Edwin Hurst. // Translation of the American Society of Civil Engineer. — 1951. — Vol. 116. — Pp. 770–799.

227. Preis T. et al. *Accelerated fluctuation analysis by graphic cards and complex pattern formation in financial markets* / Tobias Preis, Peter Virnau, Wolfgang Paul, Johannes J. Schneider. — New Journal of Physics. — 2009. — Vol. 11, № 9.

228. Schaefer A. et al. *A comparative analysis of spectral exponent estimation techniques for 1/f(β) processes with applications to the analysis of stride interval time series* / Alexander Schaefer, Jennifer S. Brach, Subashan Perera, Ervin Sejdić. — Journal of Neuroscience Methods. — 2014. — Vol. 222. — Pp. 118–130.

229. Kim B.-S. et al. *Hurst's Memory for Chaotic, Tree Ring, and SOI Series* / Byung-Sik Kim, Hung-Soo Kim, Sun-Hong Min. — Applied Mathematics. — 2014. — Vol. 5, № 1. — Pp. 175–195.

230. Kirichenko L. et al. *Comparative analysis for estimating of the Hurst exponent for stationary and nonstationary time series* / Ludmila Kirichenko, Tamara Radivilova, Zhanna Deineko. — Information Technologies & Knowledge. — 2011. — Vol. 5. — Pp. 371–388.

231. Grech D. *Impact of Scaling Range on the Effectiveness of Detrending Methods* / Dariusz Grech, Zygmunt Mazur // Acta Physica Polonica. — Vol. 127 (2015). — № 3-A. — Pp. A-59–A-65.

232. Muzy J. F. et al. *Wavelets and multifractal formalism for singular signals: Application to turbulence data* / Jean-François Muzy, Emmanuel Bacry, Alain Arneodo. — Physical Review Letters. — 1991. — Vol. 67, № 25. — Pp. 3515–3518.

233. Мандельброт Б. Б. *Фрактальная геометрия природы* / Бенуа Б. Мандельброт; пер. с англ. А. Р. Логунова — М. : Институт компьютерных исследований, 2002. — 656 с.

234. López J. L. *Performance of multifractal detrended fluctuation analysis on short time series* / Juan Luis López, Jesús Guillermo Contreras. — Physical Review E. — 2013. — Vol. 87. — № 2.

235. Hu K. et al. *Effect of Trends on Detrended Fluctuation Analysis* / Kun Hu, Plamen Ch. Ivanov, Zhi Chen, Pedro Carpena, H. Eugene Stanley. — Physical Review E. — 2001. — Vol. 64, № 1. — Pp. 011114–011133.

236. Beck C. *Thermodynamics of Chaotic Systems. An Introduction* / Christian Beck, Friedrich Schögl. — Cambridge : Cambridge University Press, 1993. — 286 p.

237. Прогонов Д. О. *Виявлення стеганограм з даними, прихованими в області перетворення цифрових зображень* / Прогонов Д. О., Кущ С. М. // Вісник НТУ України «Київський політехнічний інститут». Серія – Радіотехніка. Радіоапаратобудування. — 2014. — Вип. № 57. — С. 128–142.

238. Кроновер Р. М. *Фракталы и хаос в динамических системах. Основы теории* / Ричард М. Кроновер; пер. с англ. под ред. Т. Э. Кренкеля. — М. : «Постмаркет», 2000. — 352 с.

239. Уэлстид С. *Фракталы и вейвлеты для сжатия изображений в действии* / Стивен Уэлстид; пер. с англ. Л. В. Печникова. — М. : «Триумф», 2003. — 320 с.

240. Болотов В. Н., Ткач Ю. В. *Выделение фрактальных сигналов в условиях сложной электромагнитной обстановки* // Электромагнитные явления. — Т. 3, № 2 (10). — 2003. — С. 211–227.

241. Пащенко Р. Э., Пащенко Э. И. *Формирование кодофазоманипулируемых фрактальных сигналов на основе последовательности Морса-Туэ* // Збірник наукових праць Харківського університету Повітряних Сил. — № 3 (25). — 2010. — С. 78–82.

242. Емец Ю. В. *Оценивание параметра мультипликативного шума на изображении с помощью мультифрактальных показателей* // Искусственный интеллект. — № 4. — 2010. — С. 212–220.

243. Arneodo A. et al. *A wavelet-based method for multifractal image analysis. Part I. Methodology and test applications on isotropic and anisotropic random rough surfaces* / Alain Arnéodo, Nicolas Decoster, Stéphane G. Roux // The European Physical Journal B – Condensed Matter and Complex Systems. — 2000. — Vol. 15, № 3. — Pp. 567–600.

244. Ганченко В. и др. *Метод выделения областей на изображениях с использованием фрактальных и текстурных характеристик снимков высокого разрешения* / В. Ганченко, А. Дудкин, А. Петровский, Б. Собковяк // Искусственный интеллект. — № 3. — 2009. — С. 523–531.

245. Павлов А. Н., Анищенко В. С. *Мультифрактальный анализ сложных сигналов* // Успехи физических наук. — Т. 177, № 8. — 2007. — С. 859–876.

246. Павлов А. Н. и др. *Мультифрактальный анализ хаотической динамики взаимодействующих систем* / А. Н. Павлов, О. В. Сосновцева, А. Р. Зиганшин // Известия вузов. Сер. «Прикладная нелинейная динамика». — Т. 11, № 2. — 2003. — С. 39–54.

247. Addison P. S. *Fractals and Chaos: An illustrated course* / Paul S. Addison. — London : Institute of Physics Publishing, 1997. — 256 p.

248. Ахметханов Р. С. *Применение теории фракталов и вейвлет-анализа для выявления особенностей временных рядов при диагностике систем* // Вестник научно-технического развития – Национальная Технологическая Группа. — № 17. — 2009. — С. 26–31.

249. Кириченко Л. О., Кобицкая Ю. А. *Методы распознавания фрактальных временных рядов с помощью характеристик информационной сложности.* — Материалы VII Международной научно-практической конференции «Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій». — Запорожье : ЗНТУ, 2014. — С. 166–167.

250. Волошина О. А. и др. *Метод ЭКГ диагностики функционального состояния человека на основе фрактального анализа и вейвлет преобразования* / Волошина О. А., Олейник В. П., Кулиш С. Н., Сами А. О. // Радіоелектронні і комп'ютерні системи. — № 4 (45). — 2010. — С. 29–34.

251. Russell D. A. et al. *Dimension of strange attractors* / David A. Russell, James D. Hanson, Edward Ott. — Physical Review Letters. — Vol. 45, № 14. — 1980. — Pp. 1175–1178.

252. Klinkenberg B. *A review of methods used to determine the fractal dimension of linear features* / Brian Klinkenberg. — Mathematical Geology. — Vol. 26, № 1. — 1994. — Pp. 23–46.

253. Кириченко Л. О. *Вейвлет-оценивание показателя Херста по временным рядам со значительными циклическими компонентами с применением пакетного вейвлет-преобразования* // АСУ и приборы автоматики. — № 158. — 2012. — С. 4–12.

254. Kestener P. *A three-dimensional wavelet based multifractal method: about the need of revisiting the multifractal description of turbulence dissipation data* / Pierre Kestener, Alain Arneodo // Physical Review Letter. — 2003. — Vol. 91, № 19.

255. Jaffard S. et al. *Wavelet Leaders in Multifractal Analysis* / Stéphane Jaffard, Bruno Lashermes, Patrice Abry. — Wavelet Analysis and Applications. — Ed. Qian Tao, Vai Mang I, Xu Yuesheng. — Birkhäuser Basel, 2007. — Pp. 201–246.

256. Chaudhuri B. B. *Texture segmentation using fractal dimension* / Bidyut Baran Chaudhuri, Nirupam Sarkar. — IEEE Transactions on Pattern Analysis and Machine Intelligence. — 1995. — Vol. 17, № 1. — Pp. 72–77.

257. Li J. et al. *An Improved Box-Counting Method for image fractal dimension estimation* / Jian Li, Qian Du, Caixin Sun // Pattern Recognition. — 2009. — Vol. 42, № 11. — Pp. 2460–2469.

258. Fawcett T. *An introduction to ROC analysis* / Tom Fawcett // Pattern Recognition Letters. — Vol. 27, № 8. — 2006. — Pp. 861–874.

259. Прогонов Д. А. *Мультифрактальный флуктуационный анализ стеганограмм* / Прогонов Д. А., Кущ С. Н. // Системные исследования и информационные технологии. — 2015. — Вып. № 4. — С. 39–47.

260. Прогонов Д. А. *Спектральный анализ стеганограмм* / Прогонов Д. А., Кущ С. Н. // Научный журнал «Радиоэлектроника, информатика, управление». — 2015. — Вып. № 2 (33). — С. 71–81.

261. Kuznetsov S. P. *Hyperbolic Chaos. A Physicist's View* / Sergey P. Kuznetsov. — Berlin : Springer, 2012. — 320 p.

262. *Cisco 2015 Annual Security Report* [Internet-ресурс] / Cisco Inc. — Режим доступу: http://www.cisco.com/web/offer/gist_ty2_asset/Cisco_2015_ASR.pdf

263. *Cisco 2017 Annual Cyber security Report* [Internet-ресурс] / Cisco Inc. — Режим доступу: https://www.cisco.com/c/m/en_au/products/security/offers/annual-cybersecurity-report-2017.html

264. Aly M. *Survey on Multiclass Classification Methods* / Mohamed Aly // Neural Networks. — 2005. — Vol. 11. — Pp. 1–9.

265. Rocha A. *Multiclass From Binary: Expanding One-Versus-All, One-Versus-One and ECOC-Based Approaches* / Anderson Rocha, Siome Klein Goldenstein // IEEE Transactions on Neural Networks and Learning Systems. — Vol. 25, № 2. — 2014. — Pp. 289–302.

266. Прогонов Д. А. *Деструкция стеганограмм с данными, встроенными в области преобразования цифровых изображений* / Прогонов Д. А., Яцура Н. П., Кущ С. Н. — Зб. наук. доп. та тез науково-технічної конференції «Інформаційна безпека України». — К. : КНУ ім. Тараса Шевченка, 2015. — С. 20–21.

267. Прогонов Д. А. *Эффективность деструкции стеганограмм с применением стандартных методов обработки изображений* / Прогонов Д. А., Яцура Н. П. — Материалы V Международной научно-технической конференции ITSEC. — К. : НАУ, 2015. — С. 14–15.

268. Черепахова К. В. *Медіанна, адаптивно-медіанна та двовимірна вінерівська фільтрація стеганограм* / Черепахова К. В., Прогонов Д. О., Кущ С. М. — Матеріали XIII Всеукраїнської науково-практичної конференції «Теоретичні та прикладні проблеми фізики, математики та інформатики». — Т. 1. — К. : ВПІ ВПК «Політехніка», 2015. — С. 192–194.

269. Черепахова К. В. *Вейвлет-аналіз стеганограм* / Черепахова К. В., Прогонов Д. О., Кущ С. М. — Матеріали XIII Всеукраїнської науково-практичної конференції «Теоретичні та прикладні проблеми фізики, математики та інформатики». — Т. 1. — К. : ВПІ ВПК «Політехніка», 2015. — С. 195–197.

270. Denemark T. *Further study on the security of S-UNIWARD* / Tomáš Denemark, Jessica Fridrich, Vojtěch Holub. — Proc. SPIE 9028, Media Watermarking, Security, and Forensics. — Ed. Alattar A. M., Memon N. D., Heitzenrater C. D. — San Francisco, 2014. — Vol. 9028. — Pp. 1601–1615.

271. Birgé L. *From Model Selection to Adaptive Estimation* / Lucien Birgé, Pascal Massart // Festschrift for Lucien Le Cam. Research Papers in Probability and Statistics. — Ed.: Pollard D., Torgersen E., Yang G. L. — New York : Springer, 1997. — Pp. 55–87.

НАВЧАЛЬНЕ ВИДАННЯ

Георгій Филимонович КОНАХОВИЧ
Дмитро Олександрович ПРОГОНОВ
Олександр Юрійович ПУЗИРЕНКО

КОМП'ЮТЕРНА СТЕГАНОГРАФІЧНА ОБРОБКА Й АНАЛІЗ МУЛЬТИМЕДІЙНИХ ДАНИХ

ПІДРУЧНИК

Друкується в авторській редакції

Комп'ютерне верстання О. Ю. Пузиренко

Підписано до друку 07.03.2018. Формат 60x84 1/16.
Друк лазерний. Папір офсетний. Гарнітура Times New Roman.
Умовн. друк. арк. 32,45. Тираж 500 прим.

ТОВ «Центр учбової літератури»
вул. Лаврська, 20, м. Київ

Свідоцтво про внесення суб'єкта видавничої справи до державного реєстру
видавців, виготівників і розповсюджувачів видавничої продукції
ДК № 2458 від 30.03.2006 р.