

Вінницький національний технічний університет

Міністерство освіти і науки України

Кваліфікаційна наукова

праця на правах рукопису

ЦІХОЦЬКИЙ МИКИТА СЕРГІЙОВИЧ

УДК 004.056+004.631.54

ДИСЕРТАЦІЯ

МЕТОДИ КРИПТОГРАФІЧНОГО ЗАХИСТУ ЗОБРАЖЕНЬ

Спеціальність 125 – «Кібербезпека»

Галузь знань 12 – «Інформаційні технології»

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ М. С. Ціхоцький

Науковий керівник:

Лужецький Володимир Андрійович,
доктор технічних наук, професор

Вінниця – 2026

АНОТАЦІЯ

Ціхоцький М. С. Методи криптографічного захисту зображень. – Кваліфікаційна наукова праця на правах рукопису. Дисертація на здобуття наукового ступеня доктора філософії в галузі знань 12 Інформаційні технології за спеціальністю 125 «Кібербезпека» – Вінницький національний технічний університет, Вінниця, 2026.

Дисертаційну роботу присвячено розв’язанню науково-прикладного завдання підвищення ефективності криптографічного захисту зображень у розподілених інформаційних системах шляхом розроблення методів шифрування зображень, методів розподілу секретного вмісту зображень та програмних засобів, що реалізують ці методи. Актуальність дослідження визначається необхідністю забезпечення конфіденційності та цілісності графічних даних у системах зберігання, передавання та резервного копіювання, де недоцільним є збереження повної копії зображення на одному вузлі. Відомі методи шифрування та схеми розподілу секрету часто передбачають застосування багатораундових криптографічних перетворень, операцій у скінченних полях, поліноміальної інтерполяції, складних математичних конструкцій або формування часток із надлишковим сумарним обсягом. Для зображень, що мають значний обсяг, такі підходи можуть призводити до збільшення обчислювальних витрат, ускладнення реалізації та зростання обсягу даних, які необхідно зберігати або передавати. Це підтверджує необхідність розробки нових методів криптографічного захисту зображень, які забезпечать пришвидшення процесів шифрування та розподілу зображень за рахунок використання простого математичного апарату, та зменшення обсягу даних які необхідно передавати та зберігати.

Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. У вступі обґрунтовано актуальність теми дослідження, визначено мету роботи, основні задачі, об’єкт і предмет дослідження.

Наведено наукову новизну та практичне значення отриманих результатів, а також відомості про публікації й апробацію матеріалів дисертації.

У першому розділі проведено аналіз сучасних методів розподілу та захисту зображень. Розглянуто базову модель схеми розподілу секрету, визначено роль дилера, учасників, часток, допустимих множин доступу та процедури відновлення секрету. Проаналізовано моделі доступу, зокрема порогові схеми типу (k,n) та граничний випадок (n,n) . Розглянуто основні класи методів розподілу секрету на основі математичних перетворень, зокрема поліноміальні схеми, лінійні та афінні схеми, а також схеми на основі китайської теореми про залишки. Показано, що такі підходи забезпечують формалізовані умови доступу та відновлення, однак у задачах обробки зображень можуть потребувати значної кількості арифметичних операцій, додаткових перетворень або збільшення обсягу часток.

Також у першому розділі розглянуто методи захисту зображень на основі криптографічного шифрування. Проаналізовано застосування класичних криптографічних алгоритмів, методів на основі хаотичних систем та комбінованих підходів. Встановлено, що наявні методи здатні забезпечувати криптографічне перетворення зображень, однак часто характеризуються багатораундовою структурою, складними операціями, залежністю від параметрів хаотичних систем або надлишковим обсягом розподілених даних. За результатами аналізу обґрунтовано доцільність розроблення методів, які забезпечують пришвидшення процесів шифрування та розподілу зображень.

У другому розділі розроблено методи шифрування та розподілу секретного вмісту зображення. Запропоновано метод шифрування зображення, який поєднує побайтову підстановку та перестановку елементів у межах усього зашифрованого вектора. Підстановка забезпечує зміну значень байтів, а перестановка змінює їх позиції, що дозволяє одночасно впливати на значення елементів та їх просторову організацію. Для формування такої перестановки використано генератор

псевдовипадкової послідовності на основі регістру зсуву з лінійним зворотним зв'язком, поділу простору індексів на піддіапазони та систему лічильників, які разом генерують номер для визначення позиції елементу у межах всього вектору. Така побудова забезпечує відтворювану перестановку та дає змогу виконати точне відновлення початкової байтової послідовності за наявності секретних параметрів.

У цьому ж розділі запропоновано удосконалений метод розподілу секретного вмісту зображення за (n,n) -схемою з попереднім шифруванням зображення. Було розроблено коаліційну (k,n) -схему розподілу секретного вмісту зображення, у якій частки формуються відповідно до допустимих коаліцій учасників, а частка дилера використовується для відновлення відсутньої складової зашифрованого вектора. Розроблено також спрощений метод розподілу за пороговою (k,n) -схемою з перекриттям блоків, у якому формування часток виконується за циклічним правилом покриття блоків. Для запропонованих методів визначено властивості детермінованості, оборотності, коректності відновлення та відсутності доступу до відкритого зображення за окремими частками. Виконано оцінку обсягу часток і операційну складність реалізації процедур шифрування, розподілу та відновлення. Теоретична оцінка показала, що PS-варіант запропонованого методу шифрування потребує від 5,6 до 59,6 разів менше операцій порівняно з розглянутими відомими методами шифрування.

У третьому розділі розроблено програмні засоби реалізації запропонованих методів шифрування та розподілу зображень. Визначено структуру даних і програмних компонентів. Розроблено програмну реалізацію методу шифрування у двох варіантах: SP та PS. У варіанті SP підстановка і перестановка виконуються послідовно з формуванням проміжного результату, тоді як у варіанті PS ці операції поєднуються в одному проході, що зменшує кількість проміжних операцій і потребу в додатковому буфері. Також розроблено програмну реалізацію коаліційного методу розподілу та відновлення за (k,n) -схемою і програмну

реалізацію спрощеного методу з перекриттям блоків.

Окремо розглянуто практичні сценарії застосування запропонованих (k,n) -схем. Для коаліційної схеми запропоновано протокол анонімної передачі секретного вмісту зображення у піринговій мережі, у якому частки передаються різними маршрутами через проміжні вузли. Для спрощеної схеми запропоновано протокол розподіленого зберігання чутливих медичних даних, у якому початковий файл попередньо зашифровується, розподіляється між вузлами зберігання, а відновлення виконується за допустимою множиною часток із перевіркою їх цілісності.

У четвертому розділі проведено експериментальну перевірку запропонованих методів шифрування та розподілу секретного вмісту зображень. Дослідження статистичних властивостей показало, що після зашифрування розподіл значень байтів наближається до рівномірного, ентропія – до граничного значення для 8-бітного подання, а кореляційні залежності між сусідніми елементами істотно зменшуються. Статистичне тестування за методикою NIST SP 800-22 підтвердило відсутність виражених закономірностей у бітових послідовностях, сформованих із зашифрованих даних.

Експериментально встановлено, що PS-варіант запропонованого методу забезпечує пришвидшення повного часу шифрування приблизно у 3–21 раз порівняно з відомими методами. Для коаліційної та спрощеної (k,n) -схем підтверджено коректність відновлення зашифрованого вектора за допустимою множиною часток, а також зменшення сумарного обсягу індивідуальних часток у 1,25–5 разів для розглянутих конфігурацій. Окремо показано, що запропонований протокол анонімної передачі ускладнює встановлення зв'язку між відправником, отримувачем і окремими частками зображення.

Отримані результати підтверджують коректність, працездатність і практичну доцільність запропонованих методів шифрування та розподілу секретного вмісту

зображень для задач захищеного зберігання, передавання та відновлення даних у розподіленому середовищі.

Результати дисертаційної роботи впроваджено в ТОВ «Каскад-Безпека» у системі захисту та розподіленого зберігання графічних даних, у розробках ТОВ «ІНТЕРНЕТ-РЕКЛАМА РІА» щодо систем захищеного передавання та відмовостійкого зберігання зображень, а також у навчальний процес кафедри захисту інформації Вінницького національного технічного університету під час викладання дисциплін «Прикладна криптологія» та «Криптографічний захист інформації».

Ключові слова: криптосистема, шифрування, розшифрування, розподіл конфіденційних зображень, розподілені обчислення, група довірених користувачів, пірінгова мережа, кіберзагроза, маршрутизація, приховування даних, біомедичні зображення, JPEG, ущільнення зображень, статистичний аналіз, показник якості.

ABSTRACT

Tsikhotskyi M. S. Methods of cryptographic image protection. – Qualifying scientific work on the rights of the manuscript. Dissertation for the degree of Doctor of Philosophy in the field of knowledge 12 «Information Technologies», specialty 125 «Cybersecurity». – Vinnytsia National Technical University, Vinnytsia, 2026.

The dissertation is devoted to solving the scientific and applied problem of improving the efficiency of cryptographic protection of images in distributed information systems by developing image encryption methods, methods for sharing the secret content of images, and software tools that implement these methods. The relevance of the research is determined by the need to ensure the confidentiality and integrity of graphical data in storage, transmission, and backup systems where storing a complete copy of an image on a single node is not appropriate. Known encryption methods and secret sharing schemes often involve multi-round cryptographic transformations, finite-field operations, polynomial interpolation, complex mathematical constructions, or the formation of shares with an excessive total volume. For images characterized by a large data volume, such approaches may lead to increased computational costs, more complex implementation, and a larger amount of data that must be stored or transmitted. This confirms the need to develop new methods of cryptographic image protection that provide faster image encryption and distribution through low-complexity mathematical operations and reduce the amount of data that must be transmitted and stored.

The dissertation consists of an introduction, four chapters, conclusions, a list of references, and appendices. The introduction substantiates the relevance of the research topic and defines the aim of the work, the main research tasks, the object, and the subject of the study. It also presents the scientific novelty and practical significance of the obtained results, as well as information on publications and approbation of the dissertation materials.

The first chapter analyzes modern methods of image distribution and protection. The basic model of a secret sharing scheme is considered, and the roles of the dealer, participants, shares, admissible access sets, and the secret recovery procedure are defined. Access models are analyzed, in particular threshold schemes of the (k,n) type and the limiting case of the (n,n) scheme, in which recovery is possible only when all generated shares are available. The main classes of secret sharing methods based on mathematical transformations are considered, including polynomial schemes, linear and affine schemes, and schemes based on the Chinese Remainder Theorem. It is shown that such approaches provide formalized access and recovery conditions; however, in image processing tasks they may require a significant number of arithmetic operations, additional transformations, or an increased share volume.

First chapter also considers image protection methods based on cryptographic encryption. The application of classical cryptographic encryption algorithms, chaos-based methods, and combined approaches is analyzed, including approaches in which encryption is combined with secret sharing, digital signal processing transformations, or compressive sensing. It is established that existing methods can provide cryptographic transformation of images; however, they are often characterized by a multi-round structure, complex operations, dependence on the parameters of chaotic systems, or an excessive volume of distributed data. Based on the analysis, the feasibility of developing methods that use simple byte-wise operations, pseudorandom permutation, and threshold sharing of encrypted image content is substantiated.

The second chapter develops methods for encryption and sharing of the secret content of an image. The initial image is represented as a byte sequence, which makes it possible to perform processing independently of the graphical data format and to consider the image as an ordered byte vector. An image encryption method is proposed that combines byte-wise substitution and permutation of elements within the entire encrypted vector. Substitution changes byte values, whereas permutation changes their positions,

which makes it possible to affect both the values of elements and their spatial organization. To form such a permutation, a pseudorandom sequence generator based on a linear feedback shift register (LFSR), a division of the index space into subranges, and a system of counters are used; together, these components generate a number for determining the position of an element within the entire vector. This construction provides a reproducible permutation and enables exact recovery of the initial byte sequence when the secret parameters are available.

The same chapter proposes an improved method for sharing the secret content of an image according to the (n,n) -scheme with preliminary image encryption. A coalition-based (k,n) -scheme for sharing the secret content of an image is developed, in which shares are formed according to admissible coalitions of participants, while the dealer's share is used to recover the missing component of the encrypted vector. A simplified threshold (k,n) sharing method with block overlap is also developed, in which shares are formed according to a cyclic rule of block coverage. For the proposed methods, the properties of determinism, reversibility, correctness of recovery, and absence of access to the plaintext image from individual shares are defined. The share volume and the operational complexity of implementing encryption, sharing, and recovery procedures are evaluated. The theoretical evaluation shows that the PS variant of the proposed encryption method requires 5.6 to 59.6 times fewer operations compared with the considered known encryption methods.

The third chapter develops software tools for implementing the proposed image encryption and sharing methods. The structure of data and software components is defined. The software implementation of the encryption method is developed in two variants: SP and PS. In the SP variant, substitution and permutation are performed sequentially with the formation of an intermediate result, whereas in the PS variant these operations are combined in a single pass, which reduces the number of intermediate operations and the need for an additional buffer. The software implementation of the

coalition-based method of sharing and recovery according to the (k,n) -scheme is also developed, as well as the software implementation of the simplified method with block overlap.

Practical scenarios for applying the proposed (k,n) -schemes are considered separately. For the coalition-based scheme, a protocol for anonymous transmission of the secret content of an image in a peer-to-peer network is proposed, in which shares are transmitted along different routes through intermediate nodes. For the simplified scheme, a protocol for distributed storage of sensitive medical data is proposed, in which the initial file is preliminarily encrypted and distributed among storage nodes, while recovery is performed using an admissible set of shares with verification of their integrity.

In the fourth chapter, an experimental verification of the proposed methods for encryption and sharing of the secret content of images was carried out. The study of statistical properties showed that, after encryption, the distribution of byte values approaches a uniform distribution, the entropy approaches the limiting value for an 8-bit representation, and the correlation dependencies between neighboring elements are significantly reduced. Statistical testing according to the NIST SP 800-22 methodology confirmed the absence of pronounced regularities in the bit sequences formed from the encrypted data.

It was experimentally established that the PS variant of the proposed method provides an acceleration of the total encryption time by approximately 3–21 times compared with known methods. For the coalition-based and simplified (k,n) -schemes, the correctness of encrypted vector recovery using an admissible set of shares was confirmed, as well as a reduction in the total volume of individual shares by 1.25–5 times for the considered configurations. It was also shown separately that the proposed anonymous transmission protocol complicates the establishment of a relationship between the sender, the receiver, and individual image shares.

The obtained results confirm the correctness, operability, and practical feasibility

of the proposed methods for encryption and sharing of the secret content of images for the tasks of secure storage, transmission, and recovery of data in a distributed environment.

The results of the dissertation were implemented at LLC “КАСКАД-Безпека” in a system for the protection and distributed storage of graphical data, in the developments of LLC “ІНТЕПНЕТ-ПЕКІАМА RIA” related to systems for secure transmission and fault-tolerant storage of images, as well as in the educational process of the Department of Information Protection of Vinnytsia National Technical University during the teaching of the disciplines “Applied Cryptology” and “Cryptographic Information Protection.”

Keywords: cryptosystem, encryption, decryption, sharing confidential images, distributed computing, group of trusted users, peer-to-peer network, cyber threat, routing, data hiding, biomedical images, JPEG, image compression, statistical analysis, quality indicator.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ ТА ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЇ

Статті в журналах, що включені до переліку наукових фахових видань України:

1. Ціхоцький М.С., Лужецький В.А. Аналіз методів розподілу секрету Електронне фахове наукове видання "Кібербезпека: освіта, наука, техніка", вип. 1, вип. 25, с. 279–293, Вер 2024. DOI: <https://doi.org/10.28925/2663-4023.2024.25.279293>
2. Luzhetsky V., Tsikhotskyi M. Image encryption and distribution method based on LFSR and counters. Information Technologies and Computer Engineering, 22(3), 77-88, 2025. DOI: <https://doi.org/10.31649/vitce/3.2025.77>
3. Лужецький В.А., Ціхоцький М.С. Розподіл секретного вмісту даних за (k,n)-схемою з використанням зашифрованих блоків. Вісник Вінницького політехнічного інституту. №5, 113-120, 2025. DOI: <https://doi.org/10.31649/1997-9266-2025-182-5-113-120>

Публікації в матеріалах конференцій, тезах доповідей:

4. Лужецький В.А., Ціхоцький М.С. Схема порогового розподілення секрету « k,n » МНПК «ІТКМ-2024». 21 – 24 травень 2024.
URL: https://journal.compsc.if.ua/test/public/journals/zbirnyk_2024.pdf.
5. Лужецький В.А., Ціхоцький М.С. Протокол анонімної передачі зображень / Міжнародна науково-практична конференція "Інформаційні технології та комп'ютерне моделювання" (МНПК «ІТКМ-2024»). 227-228. 18 – 22 травень 2026. URL: <https://drive.google.com/file/d/14wz-rFrlYJyctAkV-lZ8RjykfPhT4Qqw/view>
6. Ціхоцький М.С. Метод розподілу зашифрованого зображення за (k,n) -схемою з використанням коаліцій учасників / Матеріали міжнародної науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2026)» Вінниця: ВНТУ, 2026. URL: <https://press.vntu.edu.ua/index.php/vntu/catalog/book/914>
7. Ціхоцький М. С. Протокол розподіленого зберігання медичних даних за пороговою (k,n) -схемою. ITSec: Безпека інформаційних технологій : матеріали XV Міжнар. наук.-техн. конф., м. Тернопіль, 27–29 трав. 2026 р. Тернопіль : ТНТУ ім. І. Пулюя, 2026. С. 145–146.

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	18
Вступ.....	19
1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ КРИПТОГРАФІЧНОГО ЗАХИСТУ ЗОБРАЖЕНЬ	25
1.1 Підходи до захисту зображень на основі розподілу секрету.....	25
1.1.1 Модель схеми розподілу секрету	25
1.1.2 Моделі доступу та роль дилера	26
1.2 Методи розподілу секрету на основі математичних перетворень	27
1.2.1 Поліноміальні схеми розподілу секрету.....	28
1.2.2 Лінійні та афінні схеми розподілу секрету	30
1.2.3 Схеми розподілу секрету на основі китайської теореми про залишки	32
1.3 Захист зображень методами шифрування	35
1.3.1 Застосування криптографічних алгоритмів шифрування.....	37
1.3.2 Методи шифрування зображень на основі хаотичних систем	38
1.4 Комбіновані підходи до захисту та розподілу зображень	40
1.4.1 Методи перетворення зображень на основі алгоритмів цифрової обробки сигналів	41
1.4.2 Поєднання перетворення зображення та схем розподілу	43
1.4.3 Методи на основі перетворень з ущільненням	44
1.5 Висновки до першого розділу.....	45
2 РОЗРОБКА МЕТОДІВ ШИФРУВАННЯ ТА РОЗПОДІЛУ ЗОБРАЖЕННЯ	47

2.1 Початкова обробка зображення.....	47
2.2 Метод зашифрування зображення.....	49
2.2.1 Реалізація підстановок.....	54
2.2.2 Реалізація операції перестановки	56
2.2.3 Структура секретного ключа	60
2.3 Методи розподілу секретного вмісту зображення	61
2.3.1 Метод розподілу секретного вмісту зображення за (n,n) -схемою..	61
2.3.2 Метод розподілу секретного вмісту зображення за (k,n) -схемою з використанням коаліцій учасників	63
2.3.3 Спрощений метод розподілу секретного вмісту зображення за пороговою схемою (k,n) з перекриттям блоків	67
2.4 Властивості запропонованого методу.....	71
2.4.1 Детермінованість, оборотність і коректність відновлення.....	71
2.4.2 Відсутність доступу до відкритого зображення за окремими частками	72
2.5 Оцінка обсягу часток у запропонованих (k,n) -схемах.....	74
2.5.1 Оцінка обсягу з попереднім ущільненням зображення	75
2.6 Оцінки складності реалізації методів шифрування та розподілу зображення.....	77
2.6.1 Оцінка складності реалізації зашифрування та розшифрування....	77
2.6.2 Оцінка складності реалізації методів розподілу та відновлення	81
2.8 Висновки до другого розділу	85
3 ЗАСОБИ РЕАЛІЗАЦІЇ МЕТОДІВ ШИФРУВАННЯ ТА РОЗПОДІЛУ ЗОБРАЖЕНЬ	88

3.1 Структура даних і програмних компонентів.....	88
3.2 Програмний засіб для зашифрування та розшифрування зображення	95
3.2.1 Програмна реалізація шифрування за SP варіантом	96
3.2.2 Програмна реалізація шифрування за PS варіантом	98
3.3 Програмна реалізація коаліційного методу розподілу та відновлення за (k,n)-схемою	99
3.4 Програмна реалізація спрощеного методу	103
3.5 Застосування запропонованих (k,n)-схем	105
3.5.1 Протокол анонімної передачі секретного вмісту зображення на основі коаліційної (k,n)-схеми	106
3.5.2 Протокол застосування спрощеної (k,n)-схеми для зберігання чутливих медичних даних	110
3.6 Висновки до третього розділу	115
4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ МЕТОДІВ ШИФРУВАННЯ ТА РОЗПОДІЛУ ЗОБРАЖЕНЬ	118
4.1 Дослідження статистичних властивостей методу шифрування.....	118
4.1.1 Аналіз розподілу значень та ентропії	119
4.1.2 Аналіз кореляційних залежностей	124
4.1.3 Статистичне тестування зашифрованих даних зображень.....	128
4.2 Дослідження продуктивності програмного засобу шифрування.....	130
4.2.1 Умови вимірювання та тестові дані	130
4.2.2 Порівняння варіантів SP та PS.....	132
4.2.3 Порівняння з відомими методами шифрування зображень.....	134

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CPS – схема розподілу секрету (англ., Secret sharing scheme).

КТЗ – китайська теорема про залишки (англ., Chinese remainder theorem).

РЗЛЗЗ – регістр зсуву з лінійним зворотним зв'язком (англ., Linear feedback shift register).

КСЗІ – комплексна система захисту інформації (англ., Comprehensive information protection system).

NIST SP 800-22 – набір статистичних тестів для оцінювання псевдовипадкових бітових послідовностей (англ., A statistical test suite for random and pseudorandom number generators for cryptographic applications).

SP – варіант реалізації методу шифрування з послідовним виконанням підстановки та перестановки (англ., Substitution–Permutation implementation variant).

PS – варіант реалізації методу шифрування з поєднанням підстановки та перестановки в одному проході (англ., Permutation–Substitution implementation variant).

P2P – пірингова мережа (англ., Peer-to-peer network).

DCT – дискретне косинусне перетворення (англ., Discrete cosine transform).

IWT – цілочисельне вейвлет-перетворення (англ., Integer wavelet transform).

GF – скінченне поле Галуа (англ., Galois field).

PNG – формат растрового зображення Portable Network Graphics (англ., Portable Network Graphics).

TIFF – формат растрового зображення Tagged Image File Format (англ., Tagged Image File Format).

Вступ

Актуальність роботи. У сучасних інформаційних системах зображення широко використовуються як форма подання даних для зберігання, обробки, передавання та резервного копіювання інформації. У багатьох прикладних сферах, зокрема медичній, військовій, науково-технічній та адміністративній, такі дані можуть містити конфіденційну інформацію, тому їх зберігання або передавання у відкритому вигляді створює ризик несанкціонованого доступу до повного вмісту зображення. Це зумовлює потребу в методах криптографічного захисту зображень, які забезпечують зміну відкритого представлення даних, унеможливають безпосереднє відновлення зображення без секретних параметрів і можуть бути поєднані з механізмами поділу зашифрованого вмісту на частки.

Одним із підходів до підвищення рівня захисту зображень є поєднання шифрування з розподілом секретного вмісту, за якого повні дані зображення не зберігаються в одному відкритому представленні, а відновлення можливе лише за наявності визначеної кількості часток або виконання заданих умов доступу. Такий підхід є доцільним для задач захищеного зберігання, передавання та резервного копіювання зображень, оскільки компрометація окремої частки або обмеженої кількості часток не повинна забезпечувати доступ до відкритого зображення. У цьому контексті схеми розподілу секрету можуть розглядатися як допоміжний механізм криптографічного захисту зображень, що доповнює процедуру шифрування та формалізує умови відновлення даних.

За останні роки світова наукова спільнота досягла значного прогресу в розвитку методів криптографічного захисту зображень, що охоплюють шифрування зображень, пороговий розподіл секрету та підвищення ефективності обробки даних значного обсягу. Вагомий внесок у розвиток сучасних схем розподілу секрету зробили А. Beimel (Ізраїль) [1], G. Persiano (Італія) [2], I. Cascudo (Іспанія) [3], S. Dziembowski (Польща) [4]. У напрямі розподілу секретних

зображень і прискорення процедур їх відновлення слід відзначити М. R. Holla (Індія) [5]. Методи шифрування зображень із використанням хаотичних систем, перестановок, дифузії, трансформних перетворень та статистичного аналізу зашифрованих даних досліджували S. Zia (Великобританія) [6], М. М. Hazzazi (Саудівська Аравія) [7], Т. Umar (Індія) [8]. Серед українських науковців у цьому напрямі слід відзначити В. А. Лужецького [9], В. В. Авраменка [10], С. Г. Антошук [11, 12], О. В. Онацького [13], О. В. Палагіна [14], В. П. Майданюка та О. Н. Романюка [15]. Зазначені праці відображають різні напрями досліджень, пов'язані із захистом, шифруванням, розподілом секрету, обробкою та ущільненням зображень, і підтверджують актуальність подальшого розвитку методів криптографічного захисту зображень.

Класичні схеми розподілу секрету, які часто використовуються для формалізації умов доступу до захищених даних, забезпечують строгі криптографічні властивості, проте, як правило, спираються на використання складних математичних конструкцій. Зокрема, у багатьох відомих підходах застосовуються поліноміальні перетворення, операції у скінченних полях або інтерполяційні процедури, що зумовлює значну обчислювальну складність алгоритмів. Для зображень, які є великими за обсягом масивами даних, така складність призводить до зростання обчислювальних витрат і ускладнює практичну реалізацію методів захисту у випадках, коли необхідна швидка обробка або багаторазове виконання процедур шифрування, розподілу та відновлення.

Також значна частина методів захисту зображень використовує локальну організацію обробки даних, коли перетворення виконуються над окремими блоками, фрагментами або групами пікселів. Така організація є зручною з погляду реалізації, однак вона обмежує характер перетворень межами локальних ділянок зображення. У задачах криптографічного захисту зображень доцільним є розгляд підходів, у яких обробка виконується в межах усього вектора даних, що дає змогу

змінювати не лише значення елементів, а й їх позиційні залежності у повному представленні зображення. Отже, актуальною є задача покращення існуючих підходів до криптографічного захисту зображень, а саме забезпечення пришвидшення процедур перетворень – шифрування та розподілу.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційне дослідження виконано на кафедрі захисту інформації Вінницького національного технічного університету відповідно до плану науково-дослідної роботи кафедри. Окремі результати дослідження пов'язані з науково-дослідною темою 51-КЗ «Методологія створення КСЗІ для об'єктів критичної інфраструктури», що виконувалася на кафедрі протягом 2022–2025 рр. Зв'язок із зазначеною темою полягає в розробленні методів і програмних засобів криптографічного захисту зображень, зокрема їх шифрування, розподілу секретного вмісту та відновлення за визначеними умовами доступу, що можуть використовуватися як складові механізмів захисту графічних даних у комплексних системах захисту інформації об'єктів критичної інфраструктури.

Мета і завдання дослідження. Метою дисертаційної роботи є пришвидшення процесів шифрування та розподілу секретного вмісту зображень шляхом розробки методів і засобів шифрування та схем розподілу секрету типу (n,n) та (k,n) .

Для досягнення поставленої мети розв'язано такі завдання:

- проаналізувати сучасні підходи до шифрування та розподіленого зберігання зображень;
- розробити метод шифрування зображень;
- розробити метод розподілу секретного вмісту зображень за схемою (n,n) ;
- розробити метод розподілу секретного вмісту зображень за пороговою схемою (k,n) ;
- розробити програмні засоби для розподіленого зберігання та відновлення

зображень, що реалізують запропоновані методи;

- провести експериментальні дослідження запропонованих методів.

Об'єктом дослідження є процеси шифрування зображення та розподілу секретного вмісту зображень.

Предметом дослідження є методи та засоби шифрування зображення та розподілу секретного вмісту зображень.

Методи дослідження включають загально визначену методологію виконання наукових досліджень, а саме: у дисертаційній роботі використано методи аналізу і синтезу алгоритмів обробки зображень, елементи теорії розподілу секрету, методи комбінаторного аналізу, алгоритмічні методи побудови процесу шифрування, а також методи моделювання та експериментального дослідження для оцінки властивостей запропонованих рішень.

Наукова новизна отриманих результатів

1. Вперше запропоновано метод детермінованого шифрування зображень, який на відміну від відомих методів використовує перестановку пікселів у межах усього зображення та заміну кожного пікселя, що забезпечує пришвидшення процесу шифрування.
2. Вперше запропоновано метод порогового розподілу секретного вмісту зображень за схемою (k,n) , який на відміну від відомих передбачає використання дилера для формування допоміжної складової кожної частки розподіленого зображення, що забезпечує зменшення сумарного обсягу часток, потрібного для зберігання розподіленого зображення.
3. Удосконалено метод розподіленого зберігання зображень за схемою (n,n) , який на відміну від відомих передбачає використання попередньо зашифрованого зображення, що забезпечує пришвидшення процесу розподілу секретного вмісту зображення.

Практичне значення отриманих результатів визначається можливістю їх використання у реальних розподілених системах зберігання та обробки зображень. Основні напрями практичного застосування полягають у:

1. Запропонований метод та програмний засіб, що його реалізує можуть бути використані у розподілених системах зберігання даних, де необхідно забезпечити поділ зображень між кількома вузлами без зберігання повної копії даних на жодному з них.
2. Метод придатний для обробки зображень великої роздільної здатності, оскільки його обчислювальна складність визначається лінійною складністю реалізації перестановок пікселів зображення та не містить операцій, складність яких зростає нелінійно зі збільшенням обсягу даних.
3. Розроблені алгоритми формування та відновлення часток можуть бути інтегровані у програмні системи захисту інформації, що використовуються для зберігання, передачі або резервного копіювання зображень у розподіленому середовищі.
4. Отримані результати можуть бути використані у навчальному процесі при підготовці фахівців з інформаційної безпеки та комп'ютерних наук, зокрема у курсах, присвячених захисту інформації, обробці зображень та розподіленим системам.

Результати дисертаційного дослідження впроваджено в ТОВ «Каскад-Безпека» у системі захисту та розподіленого зберігання графічних даних для захищеного зберігання, передавання та відновлення зображень у розподіленому середовищі.

Окремі результати роботи використано в розробках ТОВ «ІНТЕРНЕТ-РЕКЛАМА РІА», пов'язаних із побудовою систем захищеного анонімного передавання та відмовостійкого зберігання зображень. Зокрема, використано підходи до передавання частин зашифрованого вмісту різними маршрутами

пірингової мережі та відновлення зображення за наявності допустимої множини часток.

Також результати дисертаційної роботи впроваджено в навчальний процес кафедри захисту інформації Вінницького національного технічного університету під час проведення лекційних, лабораторних і практичних занять з дисциплін «Прикладна криптологія» та «Криптографічний захист інформації».

Особистий внесок здобувача. Основні наукові результати дисертаційної роботи отримано здобувачем особисто під час проведення досліджень у Вінницькому національному технічному університеті. У наукових працях, опублікованих у співавторстві, особисто здобувачеві належить: аналіз сучасних методів розподілу секрету, визначення їх переваг і обмежень у задачах захисту зображень [1]; розробка методу шифрування та розподілу зображення на основі РЗЛЗЗ та лічильників [2]; розробка методу розподілу секретного вмісту даних за пороговою (k,n) -схемою з використанням зашифрованих блоків [3]; пороговий розподіл секрету за (k,n) -схемою [4]; розробка протоколу анонімної передачі зображень через проміжні вузли пірингової мережі [5]; метод розподілу зашифрованого зображення за (k,n) -схемою з використанням коаліцій учасників [6]; протокол розподіленого зберігання медичних даних за пороговою (k,n) -схемою [7].

Публікації. Основні результати дисертації опубліковано в 7 роботах, з яких 3 статті у фахових виданнях та 4 тези доповідей.

Структура та обсяг дисертації. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Повний обсяг дисертації складає – 181 сторінок, у тому числі: 145 сторінка основного тексту, 46 рисунків, 11 таблиць, список використаних джерел із 115 найменувань.

1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ КРИПТОГРАФІЧНОГО ЗАХИСТУ ЗОБРАЖЕНЬ

1.1 Підходи до захисту зображень на основі розподілу секрету

У задачах захисту інформації одним із базових ризиків є централізоване зберігання даних, за якого компрометація одного вузла або носія призводить до втрати конфіденційності всього повідомлення. Для зображень, які характеризуються значним обсягом і часто містять чутливу інформацію, цей ризик є особливо критичним. У зв'язку з цим виникає задача організації зберігання даних у розподіленому вигляді, за якої жоден окремий вузол не володіє повною інформацією про збережене повідомлення.

Формально задача розподіленого зберігання полягає у побудові такого способу поділу повідомлення, за якого відновлення початкових даних можливе лише за виконання визначених умов доступу. При цьому окремі частки, що зберігаються незалежно одна від одної, не повинні містити достатньої інформації для самостійного відновлення повідомлення. Такий підхід дозволяє зменшити наслідки компрометації окремих елементів системи зберігання та задати формальні правила реконструкції даних.

Одним із загальноприйнятих інструментів формалізації зазначеної задачі є схеми розподілу секрету, які визначають математичну модель поділу повідомлення на частки та процедуру його відновлення. У контексті розподіленого зберігання зображень ці схеми розглядаються не як самодостатні криптографічні примітиви, а як базова модель, що може бути доповнена алгоритмічними методами обробки даних.

1.1.1 Модель схеми розподілу секрету

У загальному випадку схема розподілу секрету (СРС) описується такими елементами. Нехай S — початковий секрет, який підлягає розподілу. У класичному

представленні S розглядається як абстрактний об'єкт, що належить деякому простору допустимих значень $\mathbf{S}$. Подальша конкретизація секрету, зокрема у вигляді зображення, розглядається як окремий випадок цієї загальної моделі.

Процес розподілу ініціюється спеціальним суб'єктом, який називається дилером (Dealer) та позначається як D . Дилер володіє секретом S і відповідає за формування множини часток, що передаються учасникам системи. Нехай множина учасників визначається як:

$$\mathbf{P} = \{P_1, P_2, \dots, P_n\}.$$

Функція розподілу, що реалізується дилером, задається відображенням:

$$D: \mathbf{S} \rightarrow S_i, \text{ де } i = 1, 2, \dots, n,$$

у результаті якого формується набір часток $(S_1, S_2, \dots, S_n)$, де $S_i \in \mathbf{S}$ – частка, що передається учаснику P_i .

Процедура відновлення секрету описується функцією реконструкції:

$$R: S_i \rightarrow S,$$

яка відновлює початковий секрет S на основі підмножини часток, що відповідають допустимій множині учасників.

1.1.2 Моделі доступу та роль дилера

Ключовою характеристикою схем розподілу секрету є модель доступу, яка визначає, які підмножини учасників мають право на відновлення секрету. Найбільш поширеною є порогова модель типу (k, n) , у межах якої будь-яка підмножина з k або більше учасників може відновити секрет, тоді як використання меншої кількості часток не дає можливості отримати інформацію про S . Окремим граничним випадком є модель (n, n) , за якої відновлення можливе лише за наявності всіх часток.

У класичній схемі дилер відіграє центральну роль на етапі ініціалізації системи. Саме дилер визначає параметри схеми розподілу, формує частки та забезпечує їх передачу відповідним учасникам. Залежно від конкретної реалізації

дилер може розглядатися як одноразовий суб'єкт, що виконує процедуру розподілу лише на початковому етапі, або як активний елемент системи, що бере участь у повторних процедурах ініціалізації.

Важливо зазначити, що наявність дилера є типовою для багатьох класичних схем розподілу секрету та дозволяє чітко відокремити етап формування часток від етапу їх використання. У задачах розподіленого зберігання зображень така модель є зручною з точки зору реалізації, оскільки дозволяє централізовано задати параметри обробки даних і забезпечити узгодженість сформованих часток.

1.2 Методи розподілу секрету на основі математичних перетворень

У розглянутій загальній моделі розподілу секрету формування множини часток визначається відображенням розподілу, яке встановлює залежність між секретом та набором часток, що передаються учасникам схеми. Конкретна реалізація цього відображення залежить від обраного механізму формування часток і визначає як структуру отриманих даних, так і умови відновлення секрету.

У класичних схемах розподілу секрету таке відображення задається за допомогою математичних перетворень, побудованих у межах алгебраїчних структур. У цих підходах секрет та частки розглядаються як елементи відповідного простору, а їх взаємозв'язок визначається аналітично заданими залежностями. Властивості відновлення секрету, зокрема порогові умови доступу, впливають безпосередньо з властивостей використаних математичних конструкцій.

У межах цього підходу процес розподілу не задається у вигляді послідовності алгоритмічних операцій над даними, а формулюється як задача побудови залежностей між елементами математичної моделі. Це визначає як формальні гарантії коректності відновлення, так і обмеження, пов'язані зі способом подання секрету та характером виконуваних обчислень.

У даному підрозділі розглядаються основні класи методів розподілу секрету,

що використовують поліноміальні, з використанням операцій у скінченних полях та лінійно-алгебраїчні механізми. Аналіз зосереджено на принципах побудови таких схем, способах формування часток та властивостях, які визначають можливість відновлення секрету за заданими умовами доступу.

1.2.1 Поліноміальні схеми розподілу секрету

Поліноміальні схеми розподілу секрету беруть початок із класичної роботи [16], у якій секрет задається як вільний член полінома над скінченним полем, а частки формуються як значення цього полінома в різних точках. Відновлення секрету виконується за допомогою інтерполяції за не менш ніж k значеннями, що безпосередньо формалізує порогову модель (k, n) . Саме ця схема стала базовою математичною конструкцією для подальших поліноміальних підходів зокрема орієнтованих на зображення.

Модифікації поліноміальних схем розподілу секрету для зображень спрямовані насамперед на усунення викривлень, що виникають через обмеження модульної арифметики. Зокрема, у роботі [17] запропоновано стратегію заміни та перерахування для покращення якості відновлення при збереженні простої модульної арифметики. Отже, при переході від абстрактного секрету до 8-бітових піксельних даних поліноміальна СРС потребує не нової порогової моделі, а коректної адаптації до дискретного діапазону значень.

Подальший розвиток цього напрямку пов'язаний із досягненням відновлення без втрат зі збереження мінімального розміру тінювих зображень і відносно простої процедури реконструкції. У роботі [18] запропоновано СРС-схему, в якій два 8-бітові пікселі поєднуються в 16-бітове значення, а всі коефіцієнти початкового полінома використовуються для вбудовування секрету. Автор підкреслює одночасне досягнення безвратного відновлення, зменшення розміру додаткових тінювих шарів та кращу ефективність порівняно з базовою схемою.

Окремий напрям сучасних поліноміальних схем пов'язаний із підвищенням

стійкості до пошкодження часток та розширенням функціональності механізму відновлення. У роботі [19] запропоновано СРС для зображень на основі полінома Лагранжа, де частини поділяються на важливі та не важливі шари, а схема орієнтується не лише на порогове відновлення, а й на диференційовану роль часток у реконструкції. Подібна модифікація демонструє, що поліноміальна модель використовується не лише для (k,n) -розподілу, а й для складніших правил доступу, де математична база лишається тією самою, але змінюється семантика часток.

Ще один важливий напрям — зміна або гнучке налаштування параметрів доступу в межах конструкції полінома СРС. У роботі [20] розглянуто порогову схему зі змінним значенням порогу на основі біваріативного полінома, що дозволяє змінювати поріг відновлення без повторного генерування всієї схеми. Для задач захисту зображень це важливо, оскільки свідчить про перехід від жорстко фіксованої порогової структури, яка може бути незручна в певних випадках, до більш гнучких моделей доступу, однак така гнучкість досягається за рахунок ускладнення математичної побудови порівняно з класичним поліномом у схемі Шаміра.

В останніх роботах поліноміальні СРС-схеми дедалі частіше комбінуються з додатковими прикладними механізмами. У роботі [21] запропоновано схему розподілу секретного зображення на основі поліномів Лежандра, орієнтовану на безвтратне відновлення та зменшення обчислювальних витрат процедури реконструкції. У цій схемі відновлення секретного зображення виконується шляхом додавання сформованих часток у довільному порядку, що дозволяє уникнути інтерполяційної реконструкції та зменшити складність етапу відновлення. Такий підхід демонструє подальшу адаптацію поліноміальних схем до задач розподілу секретних зображень, однак зберігає загальну залежність від алгебраїчної структури формування часток. У роботі [22] СРС на основі поліномів поєднується з приховуванням інформації для отримання беззмістовних часток, тобто часток, які

не виглядають як підозрілі шумоподібні зображення. Такі праці демонструють, що сучасний розвиток поліноміальних схем відбувається у бік прикладної адаптації: підвищення рівня випадковості часток, покращення їх зовнішнього вигляду, узгодження з каналом передавання та зменшення артефактів відновлення.

Водночас розвиток поліноміальних схем не обмежується лише СРС застосуванням. У загальній теорії розподілу секрету також з'являються роботи, спрямовані на покращення безпекових властивостей самих поліноміальних конструкцій. Наприклад, [23] показали, що попередні схеми з безпечним відновленням секрету на основі біваріативних поліномів містили вразливості, і запропонували покращену архітектуру методу. Отже, навіть формально строгі поліноміальні моделі потребують ретельного аналізу не лише з позиції пороговості, а й з позиції коректності реконструкції та стійкості до активної участі стороннього противника.

Таким чином, сучасні дослідження поліноміальних схем розподілу секрету розвивають класичну ідею Шаміра у трьох основних напрямках: адаптація до піксельних даних та усунення спотворень відновлення; підвищення гнучкості моделі доступу та функціональної ролі часток; інтеграція з допоміжними механізмами, орієнтованими на практичне використання часток у реальних каналах передавання і зберігання. Разом із тим спільною для цього класу підходів залишається залежність від поліноміальної інтерполяції та операцій у скінченних полях або споріднених алгебраїчних структурах, що визначає їх математичну визначеність, але водночас задає специфічні обмеження при роботі з великими зображеннями та при необхідності мінімізувати обчислювальні витрати.

1.2.2 Лінійні та афінні схеми розподілу секрету

Окрім поліноміальних схем, у задачах розподілу секрету застосовуються підходи, що базуються на лінійних та афінних перетвореннях. У таких схемах формування часток визначається як лінійна або афінна комбінація компонентів

секрету та випадкових параметрів, заданих у відповідному алгебраїчному просторі. На відміну від поліноміальних методів, де залежність між секретом і частками задається через значення полінома, у лінійних схемах ця залежність визначається системою лінійних рівнянь або матричним відображенням.

У роботі [24] запропоновано геометричний підхід до розподілу секрету, в якому секрет визначається як точка перетину гіперплощин, а частки відповідають параметрам цих площин. Така схема є еквівалентною системі лінійних рівнянь і демонструє альтернативну до поліноміальної інтерпретацію порогового розподілу, де відновлення секрету зводиться до розв'язання системи рівнянь. Даний підхід став основою для подальшого розвитку лінійних схем, у яких структура залежностей між частками може бути задана явно через матричні представлення.

У сучасних дослідженнях лінійні схеми часто розглядаються в узагальненій формі як лінійні схеми розподілу секрету, де формування часток задається множенням вектора секрету на породжувальну матрицю. У роботі [25] систематизовано такі підходи та показано, що вони можуть описувати широкий клас схем розподілу, включаючи поліноміальні як окремий випадок. Це дозволяє розглядати лінійні схеми як загальну модель, у межах якої визначається структура доступу та залежності між частками.

Застосування лінійних схем до задач розподілу зображень зазвичай передбачає поелементне або блочне представлення зображення у вигляді векторів, після чого до них застосовується лінійне відображення. У роботі [26] запропоновано схему розподілу секретного зображення на основі лінійної алгебри, де зображення подається у вигляді блоків, а частки формуються як лінійні комбінації компонентів цих блоків із використанням параметрів схеми. Такий підхід дозволяє зберегти точність відновлення, однак передбачає виконання алгебраїчних операцій над блоками зображення, обсяг яких зростає зі збільшенням розміру вхідних даних.

У роботі [27] розглянуто використання афінної булевої класифікації у схемах розподілу секретного зображення, де формування часток пов'язується з афінними перетвореннями та прогресивним керуванням якістю відновлення. Це дозволяє підвищити варіативність сформованих часток без зміни загальної ідеї порогового відновлення, однак не усуває залежності від додаткових алгебраїчних перетворень під час формування та використання часток.

Окремий напрям розвитку пов'язаний із поєднанням схем розподілу секретних зображень із механізмами стійкості до пошкодження часток. У роботі [28] запропоновано робастну порогову схему розподілу секретного зображення, орієнтовану на відновлення секрету за наявності шуму в тіньових зображеннях. Такий підхід розширює функціональні можливості класичних схем, але потребує додаткового врахування помилок у частках під час процедури відновлення.

У роботах останніх років також розглядаються модифікації лінійних схем із використанням блочних кодів [29], де структура часток узгоджується з кодовими конструкціями, що забезпечують додаткову стійкість до атак або втрат інформації. Це свідчить про тенденцію до інтеграції лінійних моделей розподілу з іншими інструментами обробки даних.

Таким чином, лінійні та афінні схеми розподілу секрету характеризуються використанням явно заданих алгебраїчних залежностей між секретом і частками у вигляді систем рівнянь або матричних перетворень. Їхньою перевагою є узагальненість і можливість гнучкого задання структури доступу, тоді як основним обмеженням виступає залежність від розв'язання систем рівнянь при відновленні секрету та зростання обчислювальних витрат при обробці великих обсягів даних, зокрема зображень.

1.2.3 Схеми розподілу секрету на основі китайської теореми про залишки

Окремий клас методів розподілу секрету базується на використанні китайської теореми про залишки (КТЗ), відповідно до якої ціле число може бути

однозначно відновлене за набором залишків за попарно взаємно простими модулями. У межах таких схем секрет представляється як числове значення або вектор значень, а частки формуються як залишки від ділення на набір модулів [30]. На відміну від поліноміальних схем, де відновлення секрету базується на інтерполяції, у КТЗ-схемах реконструкція здійснюється шляхом обчислення за відомою системою конгруенцій. Це дозволяє уникнути використання поліноміальних операцій та зводить процес відновлення до послідовності арифметичних дій у кільці цілих чисел.

Базовим підходом у цьому напрямі є схема Asmuth–Bloom [30], у якій секрет маскується додаванням випадкової величини та далі представляється у вигляді системи залишків. Такий підхід забезпечує порогову властивість за рахунок вибору параметрів модулів і дозволяє формувати частки як компактні залишки, що не містять прямої інформації про секрет при їх окремому розгляді.

У подальших роботах КТЗ-підхід був розширений і адаптований до задач розподілу зображень. У роботі [31] запропоновано схему прогресивного розподілу секретного зображення, у якій китайська теорема про залишки поєднується з поліноміальним підходом, а у роботі [32] КТЗ використано для побудови схеми оборотного приховування даних у зашифрованій області на основі розподілу секретного зображення. Такі підходи дозволяють поєднувати порогове відновлення з додатковими функціональними можливостями, однак потребують додаткових процедур формування, обробки або вилучення службових даних.

Також частина досліджень спрямована на підвищення практичності та ефективності КТЗ-схем. У роботі [33] запропоновано ефективну систему багатосекретного розподілу кольорових зображень на основі китайської теореми про залишки та операції XOR з апаратною FPGA-реалізацією. У роботі [34] розглянуто практичну схему розподілу секретного зображення на основі КТЗ, орієнтовану на усунення втрат відновлення та обмежень порогу, характерних для

застосування алгоритму Asmuth–Bloom до зображень. Такі підходи демонструють можливість адаптації параметрів КТЗ-схем до особливостей графічних даних, однак ускладнюють загальну структуру розподілу порівняно з базовою моделлю.

Окремий напрям пов'язаний із поєднанням КТЗ із додатковими механізмами обробки даних. У роботі [35] запропоновано підхід до розподілу кількох кольорових зображень із використанням китайської теореми про залишки, де додаткові операції над піксельними даними застосовуються для формування багатосекретної структури часток. У таких схемах КТЗ виконує функцію формування та відновлення часток, тоді як допоміжні операції над даними впливають на структуру проміжного представлення зображень і ускладнюють аналіз окремих часток.

У останніх роботах також розглядається інтеграція КТЗ із методами ущільнення або перетворення сигналів. Це дозволяє поєднувати процеси зменшення обсягу даних і їх розподілу, що є актуальним для обробки зображень великого розміру. Водночас такі підходи ускладнюють загальну структуру алгоритму та вимагають додаткової синхронізації між етапами обробки.

Таким чином, КТЗ-схеми розподілу секрету побудовані на використанні модульної арифметики для формування часток і відновлення секрету без застосування поліноміальної інтерполяції. Їх перевагою є відносна простота арифметичних операцій та можливість компактного представлення часток. Разом із тим, основними обмеженнями є необхідність узгодження діапазону значень із системою модулів, складність вибору параметрів та залежність ефективності від структури вхідних даних, що є критичним у задачах розподілу зображень.

У порівнянні з поліноміальними схемами розподілу секрету, КТЗ-підходи реалізують інший механізм формування часток: замість побудови функціональної залежності у вигляді полінома вони використовують представлення секрету як системи залишків за різними модулями. У цьому випадку відновлення секрету не

потребує інтерполяції, а зводиться до обчислення за системою конгруенцій, що визначає інший характер обчислювального процесу.

Водночас поліноміальні та КТЗ-схеми мають спільну особливість: формування часток у них визначається наперед заданими математичними залежностями між секретом, параметрами схеми та отриманими частками. У таких підходах основна увага зосереджується на виконанні умов доступу та коректності відновлення. Це забезпечує формальну визначеність процедур розподілу та відновлення, однак потребує узгодження параметрів схеми з обсягом, розрядністю та способом подання даних, що обробляються.

Отже, як поліноміальні, так і КТЗ-схеми реалізують формування часток S через задані математичні залежності між секретом і частками. У подальшому доцільно розглянути підходи, у яких основна увага приділяється не лише математичній моделі доступу, а й процедурі перетворення та розподілу самого вмісту зображення, поданого у вигляді впорядкованої послідовності даних.

1.3 Захист зображень методами шифрування

Окрім підходів, що базуються на формуванні часток відповідно до схем розподілу секрету, у задачах захисту зображень широко використовуються методи, спрямовані на перетворення самого представлення даних [36, 37]. У цих підходах зображення розглядається як масив значень, до якого застосовується певне відображення, що змінює його структуру або значення елементів.

Нехай E — відображення, визначене на множині можливих представлень зображення. У загальному випадку результатом такого перетворення є нове представлення $V' = E(V)$, яке може використовуватись як самостійний результат обробки або як вхід для подальших процедур. Залежно від характеру відображення E , такі методи можуть реалізовувати як криптографічне шифрування, так і інші види перетворень, що змінюють статистичні або структурні властивості даних.

Таким чином, на відміну від схем розподілу, що визначають правила формування та відновлення часток, методи цього класу спрямовані на зміну самого подання зображення. У подальшому розглядаються основні підходи до побудови таких перетворень.

У попередньому підрозділі розглянуто підходи до розподілу секрету, в яких формування часток задається аналітично у межах математичної моделі. Такі схеми визначають умови доступу до даних, однак не передбачають зміни структури самого секрету, який у задачах обробки зображень представляється у вигляді масиву значень.

У сучасних дослідженнях щодо захисту зображень значна увага приділяється методам, що виконують перетворення даних незалежно від етапу розподілу. У літературі такі підходи, як правило, позначаються як методи шифрування зображень. У межах цих методів зображення розглядається як сукупність елементів, над якими виконується визначене перетворення з метою зміни їх структурних та статистичних характеристик.

На відміну від механізмів формування часток, де залежності між даними задаються у вигляді математичних співвідношень, у даному випадку процес обробки визначається способом побудови перетворення. Зокрема, такі методи можуть базуватися на використанні класичних криптографічних алгоритмів, динамічних систем, перетворень представлення даних або їх комбінацій. Незважаючи на різницю у реалізації, усі вони виконують спільну функцію — змінюють структуру даних без явного задання умов їх подальшого розподілу.

У рамках даного підрозділу розглядаються основні класи механізмів перетворення зображень, що використовуються у задачах захисту даних, зокрема застосування класичних криптографічних алгоритмів, методи на основі хаотичних систем, підходи, що використовують перетворення представлення зображення, а також комбіновані методи. Аналіз спрямовано на виявлення особливостей цих

підходів та їх ролі у побудові систем захисту зображень.

Таким чином, у загальному процесі обробки зображення механізми перетворення виконують функцію E , тоді як розглянуті раніше схеми розподілу визначають механізм S . Їх узгоджене застосування формує основу для побудови комплексних підходів до захисту та розподілу зображень.

1.3.1 Застосування криптографічних алгоритмів шифрування

Одним із базових підходів до захисту зображень є застосування класичних криптографічних алгоритмів, розроблених для шифрування даних загального призначення. У цьому випадку зображення розглядається як послідовність байтів, до якої безпосередньо застосовується криптографічне перетворення без урахування внутрішньої структури даних.

Найбільш поширеним представником цього класу є шифр AES [38], який має фіксований розмір блоку та визначену кількість раундів перетворення. У задачах обробки зображень AES застосовується шляхом подання зображення у вигляді лінійного байтового потоку, що дозволяє використовувати стандартні режими роботи, такі як ECB, CBC або CTR.

У роботі [39] розглянуто застосування AES для шифрування зображень із використанням різних режимів роботи, де показано, що навіть стандартні криптографічні алгоритми можуть забезпечити достатній рівень захисту за умови коректного вибору режиму. Зокрема, застосування режимів із зворотним зв'язком дозволяє уникнути появи структурних артефактів, характерних для режиму ECB.

У роботі [40] авторами було досліджено ефективність AES у задачах захисту мультимедійних даних, включаючи зображення, з точки зору швидкодії та стійкості до атак. Результати показують, що AES забезпечує високий рівень криптографічної безпеки, однак його ефективність значною мірою залежить від способу організації вхідного потоку даних.

Окремий напрям досліджень пов'язаний із оптимізацією застосування AES

для великих обсягів даних. У роботі [41] запропоновано підходи до паралельної реалізації AES для обробки зображень, що дозволяє зменшити час шифрування при збереженні рівня безпеки. Такі рішення є актуальними для систем, де необхідна обробка високороздільних зображень у реальному часі.

У роботі [42] запропоновано комбінований підхід, у якому перед застосуванням AES виконується попереднє перетворення зображення, що дозволяє підвищити ефективність шифрування та зменшити кореляцію між сусідніми пікселями. Це свідчить про те, що класичні криптографічні алгоритми часто використовуються у поєднанні з іншими методами обробки даних.

Таким чином, застосування AES та інших класичних криптографічних алгоритмів у задачах шифрування зображень забезпечує високий рівень криптографічної стійкості та формально обґрунтовану модель безпеки. Водночас такі підходи не враховують специфіку структури зображень і розглядають їх як однорідний потік даних. Це обмежує їх ефективність у задачах, де важливим є врахування просторових залежностей між елементами або інтеграція з механізмами розподілу даних.

1.3.2 Методи шифрування зображень на основі хаотичних систем

Методи шифрування зображень на основі хаотичних систем реалізують перетворення E шляхом керування обробкою даних через нелінійні динамічні моделі. Використовуючи властивості чутливості до початкових умов, топологічного перемішування та ергодичності, що дозволяє генерувати керуючі послідовності з високою складністю. У більшості робіт хаотична система не є «шифром» у класичному сенсі, а виступає джерелом керування для операцій над зображенням.

У одновимірних хаотичних системах (логістична, тенто-подібна та інші види хаотичних мап) хаотична послідовність використовується для формування порядку доступу до пікселів і параметрів їх модифікації. У роботах [43, 44] показано, що

навіть прості 1D-відображення можуть забезпечувати суттєве зниження кореляції між пікселями та зростання ентропії зашифрованих даних. Разом із тим у цих підходах спостерігається обмежений діапазон параметрів і підвищена чутливість до числових ефектів реалізації.

Подальші дослідження, як от [45, 46] спрямовані на модифікацію 1D-відображень шляхом введення частково-нелінійних або поліноміальних компонентів, що розширює простір параметрів і покращує статистичні характеристики генерованих послідовностей.

Для подолання обмежень одновимірних хаотичних відображень у методах шифрування зображень використовують системи з кількома змінними стану, зокрема хаотичні системи [47], [48] та [49]. На відміну від одновимірних відображень, такі системи формують складнішу динаміку та можуть використовуватися для отримання кількох взаємопов'язаних керувальних послідовностей, що застосовуються під час перестановки, дифузії або комбінованого перетворення елементів зображення.

Для підвищення складності керувальних послідовностей у сучасних методах шифрування зображень використовують багатовимірні хаотичні та гіперхаотичні системи. У роботі [49] запропоновано схему шифрування зображень на основі покращеної чотиривимірної хаотичної системи та еволюційних операторів, а у роботі [50] — метод шифрування з використанням двовимірної гіперхаотичної карти та динамічного DNA-Zigzag-кодування. Такі підходи дають змогу формувати складніші псевдовипадкові послідовності для перестановки, дифузії та комбінованого перетворення елементів зображення.

У роботах [51, 52] запропоновано хаотичні системи, які застосовуються у багатокрокових алгоритмах шифрування зображень. У таких схемах сформовані хаотичні послідовності використовуються для керування окремими етапами перетворення, зокрема перестановкою, дифузією або заміною значень елементів.

Це підвищує складність побудови алгоритму та розширює простір параметрів, однак водночас ускладнює вибір початкових умов і параметрів системи, що може впливати на стабільність програмної реалізації.

Значна частина сучасних алгоритмів реалізує послідовність етапів, у якій хаотичні послідовності керують: порядком обробки елементів, параметрами зміни значень, структурою доступу до блоків зображення.

У роботах [53], [54], та [55] показано, що поєднання декількох етапів дозволяє досягати кращих значень показників зміни пікселів та ентропії. Водночас ефективність таких схем значною мірою залежить від узгодженості між етапами, що не завжди формалізується у вигляді єдиної моделі.

Окремий напрям становлять методи, у яких хаотичні системи поєднуються з іншими механізмами: перетвореннями частот (DCT, DWT), методами машинного навчання, криптографічними примітивами.

У роботах [56, 57] показано, що такі поєднання дозволяють покращити статистичні характеристики та адаптивність алгоритмів. Однак це призводить до ускладнення структури та зменшує прозорість аналізу безпеки.

Незважаючи на широке застосування, у низці робіт [58, 59] показано, що хаотичні схеми можуть бути вразливими до атак із відомим відкритим текстом або до методів відновлення параметрів системи. Крім того, відсутність уніфікованої криптографічної моделі безпеки ускладнює формальну оцінку стійкості таких алгоритмів.

1.4 Комбіновані підходи до захисту та розподілу зображень

У попередніх підрозділах розглянуто два напрями розвитку підходів до захисту зображень: формування часток S та шифрування E . У більшості класичних підходів ці механізми розглядаються окремо: шифрування застосовується до зображення як самостійна процедура, а розподіл виконується над вже сформованим

масивом даних.

У дослідженнях з'являється окремий напрям, у якому ці механізми поєднуються в межах однієї схеми. У таких підходах перетворення даних впливає на процес формування часток або виконується безпосередньо в межах схеми розподілу. Це приводить до часткової інтеграції функцій E і S , де зміна представлення даних і їх розподіл розглядаються як взаємопов'язані етапи.

1.4.1 Методи перетворення зображень на основі алгоритмів цифрової обробки сигналів

Розподіл S виконується над коефіцієнтами Y , отриманими в результаті попередньої обробки зображення:

$$I \xrightarrow{\tau} Y,$$

де τ — визначене перетворення сигналу (наприклад, дискретне косинусне або вейвлет-перетворення). У результаті формується масив коефіцієнтів, який однозначно визначає вихідне зображення.

Подальша обробка полягає у застосуванні операцій до елементів Y , тобто змінюється не безпосередньо значення пікселів, а параметри, з яких ці значення відновлюються.

Методи на основі дискретного косинусного перетворення у сучасних роботах використовуються переважно у двох варіантах, що включають обробку всіх коефіцієнтів або обробку підмножини коефіцієнтів. У роботах [60, 61, 62] алгоритм будується як послідовність операцій, що включає розбиття зображення на блоки, обчислення коефіцієнтів та застосування перетворення до вибраних елементів Y . Відмінності між роботами визначаються критерієм вибору коефіцієнтів, який може бути позиційним, частотним або адаптивним. У роботі [63] показано, що різні критерії вибору призводять до різного розподілу змін у масиві коефіцієнтів, тоді як у роботі [64] використано комбінований підхід, де частина коефіцієнтів обробляється інтенсивніше, ніж інші, що змінює структуру результату без повної

обробки всього масиву.

У методах на основі вейвлет-перетворення зображення розкладається на піддіапазони, які обробляються незалежно. У роботах [65, 66, 67] перетворення E_T застосовується або до всіх піддіапазонів, або до окремих, зокрема допоміжних компонентів. Ключовим фактором є розподіл обробки між піддіапазонами: у випадку рівномірної обробки зміни охоплюють весь масив Y , тоді як нерівномірна обробка призводить до того, що частина коефіцієнтів залишається незмінною. У роботі [65] показано, що навіть при однаковому перетворенні результат залежить від того, які саме піддіапазони обробляються.

У підходах, що базуються на вибірковій обробці коефіцієнтів, основний акцент робиться не на типі перетворення τ , а на формуванні множини індексів:

$$\Omega \subset Y,$$

для яких застосовується перетворення. Алгоритм визначається способом побудови множини Ω та видом операцій над її елементами. У роботах [68, 69] використовуються різні підходи до адаптивного вибору індексів, що змінюється залежно від значень коефіцієнтів, тоді як у [70] застосовується вибіркове шифрування наперед визначених складових стисненого DCT-представлення, зокрема окремих груп DC-коефіцієнтів, AC-значень або AC-послідовностей. Таким чином, правило формування Ω визначає, які частини проміжного представлення зображення підлягають перетворенню, а характер цього правила впливає на розподіл змін у зашифрованих даних.

У комбінованих підходах [71, 72] перетворення представлення поєднується з додатковими механізмами керування. У таких схемах T визначає структуру даних, тоді як зовнішній механізм задає порядок або параметри обробки коефіцієнтів. Це призводить до збільшення кількості параметрів, появи багатомодульних схем та залежності результату від взаємодії компонентів алгоритму.

Узагальнюючи, методи цього класу характеризуються тим, що коефіцієнти Y

є результатом формального перетворення τ , а перетворення E_τ застосовується до елементів Y . Результат визначається вибором τ , формуванням множини коефіцієнтів та способом їх обробки. Таким чином, ключовим фактором є не лише саме перетворення, а й структура доступу до елементів представлення, яка визначає характер змін у даних.

1.4.2 Поєднання перетворення зображення та схем розподілу

У таких схемах зображення спочатку переводиться у зашифроване або іншим чином змінене представлення, після чого над цими даними виконується формування часток, або ж елементи шифрування вбудовуються безпосередньо у процедуру їх побудови.

Характерним прикладом такого підходу є схема запропонована у роботі [73]. Тут зображення спочатку шифрується, а далі формується набір часток без піксельного розширення та зі зменшеним розміром часток. Автори прямо розглядають шифрування як попередній етап, а СРС — як механізм подальшого розподілу вже перетворених даних. Тобто розподіл виконується не над початковими значеннями пікселів, а над їх зашифрованим представленням, що змінює властивості отриманих часток порівняно з класичними СРС.

Інший варіант реалізовано у роботі [74]. У цій схемі секретні зображення спочатку квантуються та шифруються шляхом використання механізму “самошифрування”, після чого зашифровані зображення змішуються у частки за допомогою змішаного шифрування. Тут шифрування вже не є просто окремим зовнішнім етапом, а входить до самої процедури формування часток. Крім того, автори підкреслюють низьке обчислювальне навантаження, оскільки схема переважно базується на операціях циклічного зсуву та XOR.

Ще один напрям представлений роботою [75]. У цьому випадку використовується гіперхаотичне керування інтенсивністю дифузії залежно від особливостей зображення, а процедура побудови часток безпосередньо керується

отриманими значеннями з гіперхаотичної системи. Тобто шифрування не просто передує поділу, а визначає спосіб його виконання.

Робота [76] належить до суміжного підкласу, де поєднання реалізується через техніку ущільненого зондування. У цій схемі паралельне ущільнене зондування поєднується зі схемою розподілу, тобто перетворення даних та їх поділ розглядаються спільно як частини одного алгоритму.

1.4.3 Методи на основі перетворень з ущільненням

У підходах на основі перетворень з ущільненням зображення розглядається як числовий масив x , який перетворюється у нове представлення меншої розмірності за допомогою лінійного відображення:

$$y = \Phi \cdot x,$$

де Φ — задана матриця перетворення, а y — новий набір значень, що формується як лінійні комбінації елементів x .

Таким чином, кожен елемент y не відповідає окремому пікселю, а містить інформацію про групу значень початкового зображення. У результаті формується відображення, в якому інформація розподілена по всьому масиву y , а не локалізована у конкретних позиціях.

У роботах цей підхід використовується у задачах захисту зображень у двох основних варіантах. У першому випадку нове представлення y розглядається як результат перетворення, що ускладнює безпосереднє відновлення початкових даних без знання матриці Φ . У роботах [77, 78] показано, що зміна параметрів матриці перетворення призводить до зміни всього набору значень y , оскільки кожен елемент залежить від багатьох елементів x .

У другому варіанті представлення y використовується як основа для подальшого розподілу. У роботах [79, 80] значення y розбиваються на частини, які розглядаються як частки. У цьому випадку кожна частка містить не окремі пікселі, а комбінації значень усього зображення. Відновлення можливе лише при

об'єднанні достатньої кількості таких частин, оскільки кожна з них містить лише частину інформації про початковий масив.

Особливістю таких підходів є те, що перетворення і розподіл не є повністю незалежними. Оскільки кожен елемент у формується з урахуванням багатьох елементів x , подальший поділ у автоматично приводить до розподілу інформації по частках. Таким чином, структура розподілу визначається не лише способом поділу даних, але й самим перетворенням.

У роботі [81] показано, що зміна способу формування матриці Φ впливає на характер розподілу значень y , а отже і на структуру часток. Це означає, що параметри перетворення визначають не тільки представлення даних, але й властивості їх поділу.

Таким чином, підходи на основі ущільнення відрізняються тим, що перетворення зображення одразу формує відображення, придатне для розподілу. На відміну від класичних схем, де E і S застосовуються послідовно, у цьому випадку перетворення визначає, як саме інформація буде розподілена між частками.

1.5 Висновки до першого розділу

Аналіз сучасних методів розподілу секрету та захисту даних показує, що більшість підходів орієнтовані на вирішення окремих класів задач. Класичні схеми розподілу секрету розглядають абстрактний секрет як неподільний об'єкт і зосереджуються на забезпеченні порогових властивостей доступу, не виконуючи попередніх перетворень даних. У таких схемах питання захисту інформації покладається на сам механізм розподілу та математичні властивості використаної конструкції.

Водночас у роботах, присвячених шифруванню зображень, основна увага приділяється перетворенню структури та значень пікселів із метою ускладнення несанкціонованого відновлення. У цьому випадку зображення розглядається як

цілісний об'єкт, а задача розподілу зазвичай не є предметом дослідження. Такі підходи є доцільними для сценаріїв захищеного зберігання або передавання, однак не формують модель доступу до частин даних.

У межах задач розподіленого зберігання зображень ці підходи можуть розглядатися як взаємодоповнювальні, але не взаємозамінні. Зображення як об'єкт розподілу накладає додаткові вимоги, пов'язані з великим обсягом даних, наявністю просторової та статистичної структури, а також необхідністю точного відновлення без похибок. Методи, ефективні для абстрактних секретів малої розмірності, не завжди придатні для таких умов без суттєвих модифікацій або зростання надлишковості.

У цьому контексті доцільним є розгляд підходів, у яких перед розподілом дані проходять детерміноване перетворення, що ускладнює відновлення вихідної інформації без знання секретних параметрів. Такий підхід дозволяє розглядати процедури захисту та розподілу як послідовні етапи єдиної схеми, не змінюючи базові принципи порогового доступу, але адаптуючи їх до роботи з великими та структурованими даними.

У зв'язку з цим постає задача розробки підходу, який забезпечує необхідний рівень криптографічної стійкості розподілених часток зображення. У межах цієї задачі розглядається застосування попереднього шифрування вхідного зображення з подальшим розподілом даних із використанням базової схеми типу (n,n) та порогової схеми типу (k,n) . При цьому до алгоритму шифрування висуваються вимоги обчислювальної простоти, що є важливим для обробки зображень значного обсягу. Окремою вимогою до порогової схеми розподілу є забезпечення зменшеного обсягу часток, які передаються та зберігаються між учасниками схеми.

2 РОЗРОБКА МЕТОДІВ ШИФРУВАННЯ ТА РОЗПОДІЛУ ЗОБРАЖЕННЯ

2.1 Початкова обробка зображення

Зображення може розглядатися на декількох рівнях представлення. На рівні візуальної моделі воно описується як дискретний двовимірний масив елементів, кожен з яких відповідає певному просторовому положенню та містить значення яскравості або компонент кольору. Такий підхід є типовим для методів цифрової обробки зображень, у яких операції виконуються над пікселями або коефіцієнтами, отриманими в результаті перетворення представлення даних [82].

Водночас у системах зберігання та передавання зображення зазвичай має форму файла, структура якого визначається відповідним форматом. Файл зображення може містити службові заголовки, сигнатури, таблиці параметрів, метадані, ущільнені або неущільнені області даних, інформацію про колірну модель, порядок байтів, прозорість та інші елементи. Наприклад, JPEG використовує форматне подання, пов'язане з дискретним косинусним перетворенням, квантуванням і ентропійним кодуванням [83]. PNG має визначену файлову структуру з сигнатурою та послідовністю блоків даних [84], а TIFF організовується через заголовок файла, директорії зображень і пов'язані з ними області даних [85].

У задачах захисту зображень можливі два основні рівні початкової обробки. Перший рівень пов'язаний із декодуванням файла до піксельного представлення. У цьому випадку зображення подається у вигляді матриці пікселів або послідовності компонент кольору, а подальші операції виконуються над отриманим внутрішнім поданням.

Другий рівень початкової обробки полягає у використанні файлового представлення. У цьому випадку зображення розглядається як файл, що містить

упорядковану послідовність байтів. Такий підхід узгоджується із задачею розподіленого зберігання, оскільки об'єктом захисту є повний вміст файла, а результатом коректного відновлення має бути початковий файл зображення без втрати або зміни його байтів.

У подальших дослідженнях розглядається саме файловий рівень подання, що дає змогу виконувати операції над єдиною байтовою послідовністю незалежно від особливостей внутрішньої будови конкретного формату.

Нехай вхідний файл зображення позначається через F . На етапі початкової обробки файл зчитується як скінченна впорядкована послідовність байтів:

$$F = (b_0, b_1, \dots, b_{N-1}),$$

де N – розмір файла у байтах, b_i – значення i -го байту файла.

Використання файлового байтового подання забезпечує уніфіковану початкову обробку для різних форматів зображень. Оскільки вхідні дані приводяться до послідовності байтів, подальші операції можуть виконуватися за однаковим алгоритмічним правилом для JPEG, PNG, BMP, TIFF та інших форматів. Це зменшує залежність реалізації від бібліотек декодування, особливостей внутрішньої структури окремого формату та зменшує накладні витрати.

Практична перевага такого підходу полягає у збереженні точності відновлення на рівні файла. Після виконання повного циклу, що включає зашифрування, формування часток, відновлення, розшифрування та збереження результату, має бути відновлена початкова байтова послідовність. Якщо через F' позначити файл, отриманий після завершення всіх зворотних процедур, то умовою коректності є $F' = F$.

Ця рівність визначає повну тотожність початкового та відновленого файлів на рівні байтів. Відповідно, у запропонованій моделі результат відновлення оцінюється за точним збігом файлового вмісту, а не за візуальним відображенням зображення.

Водночас файловий рівень обробки визначає певні межі застосовності методу. Зашифрована байтова послідовність може не відповідати вимогам початкового графічного формату. Зміна службових байтів, сигнатур, таблиць, маркерів або ущільнених областей даних може порушити форматну узгодженість файла. Тому проміжний результат зашифрування доцільно розглядати як захищений байтовий контейнер для подальшого зберігання або розподілу, а не як файл зображення, призначений для безпосереднього перегляду стандартними програмними засобами.

Таким чином, початкова обробка зображення зводиться до формування байтового подання файла, яке використовується як вхідний об'єкт для подальших процедур. Такий підхід забезпечує єдину модель даних для різних форматів зображень, зменшує залежність від особливостей внутрішньої будови різних форматів, спрощує інтеграцію із процедурами розподілу та задає строгий критерій коректності у вигляді повної байтової тотожності початкового й відновленого файлів.

2.2 Метод зашифрування зображення

Шенон у своїй роботі [86] започаткував спосіб представлення процедури шифрування як процес виконання двох послідовних перетворень, а саме – перестановка (diffusion) та підстановка (confusion). Також загальноприйнятим є такий підхід до зашифрування даних, коли вони розбиваються на певні блоки і при цьому операція перестановки виконується лише в межах кожного блоку. Натомість пропонується криптосистема, особливість якої полягає у виконанні перестановок елементів зображення не в межах окремих блоків, а у межах усього зображення. Таким чином досягається вища стійкість до зламу або аналізу зашифрованого секрету, за рахунок збільшення кількості можливих варіантів перестановок пікселів. Сукупність варіантів при такому підході значно перевищує варіанти

перестановок в межах одного блоку даних.

Вхідними даними для процесу зашифрування є файл зображення F , поданий у вигляді вектора $\mathbf{V}$ з послідовно зчитаних байтів відкритого файлу. Нехай $\mathbf{V} = \{v_0, v_1, \dots, v_{N-1}\}$ – вхідний вектор зображення, $\mathbf{V}' = \{v'_0, v'_1, \dots, v'_{N-1}\}$ – елементи проміжного вектору після виконання операцій підстановки або перестановки, а $\mathbf{C} = \{c_0, c_1, \dots, c_{N-1}\}$ – зашифрований вектор тієї самої довжини.

Процедура зашифрування задається функцією:

$$E_K: \mathbf{V} \rightarrow \mathbf{C},$$

де K — секретний ключ, який складається з трьох компонент K_1, K_2, K_3 , що визначають параметри підстановки, та правила формування перестановки.

Пропонується такий метод шифрування зображення, який в подальшому будемо позначати як метод SP , це означає, що для кожного байту спочатку виконується підстанова $\mathbf{V}' = S_{K_2}^{SP}(\mathbf{V})$, а потім перестановка за правилом:

$$\pi_{заш} = \begin{pmatrix} 0, 1, \dots, N-1 \\ p_0, p_1, \dots, p_{N-1} \end{pmatrix}.$$

Отже, функція зашифрування має вигляд:

$$E_K^{SP}(\mathbf{V}) = P_{K_1, K_3}(S_{K_2}^{SP}(\mathbf{V})) = \mathbf{C}.$$

Схема процесу з такою послідовністю операцій наведена на рис. 2.1.

Для розшифрування реалізується зворотна послідовність операцій. Однак, з файлу зашифрованого зображення зчитуються не окремі байти, а їх блоки, з яких формується масив зашифрованого зображення. Далі з цього масиву зчитуються байти в порядку, що відповідає оберненому правилу перестановки:

$$\pi_{розш} = \begin{pmatrix} p_0, p_1, \dots, p_{N-1} \\ 0, 1, \dots, N-1 \end{pmatrix}.$$

При цьому до кожного байту застосовується правило оберненої підстановки:

$$\mathbf{V} = S_{K_2}^{-1}(\mathbf{V}').$$

Таким чином функція розшифрування описується як:

$$D_K^{SP}(\mathbf{C}) = S_{K_2}^{-1}(P_{K_1, K_3}^{-1}(\mathbf{C})) = \mathbf{V}.$$

Схему процесу розшифрування наведено на рис. 2.2.

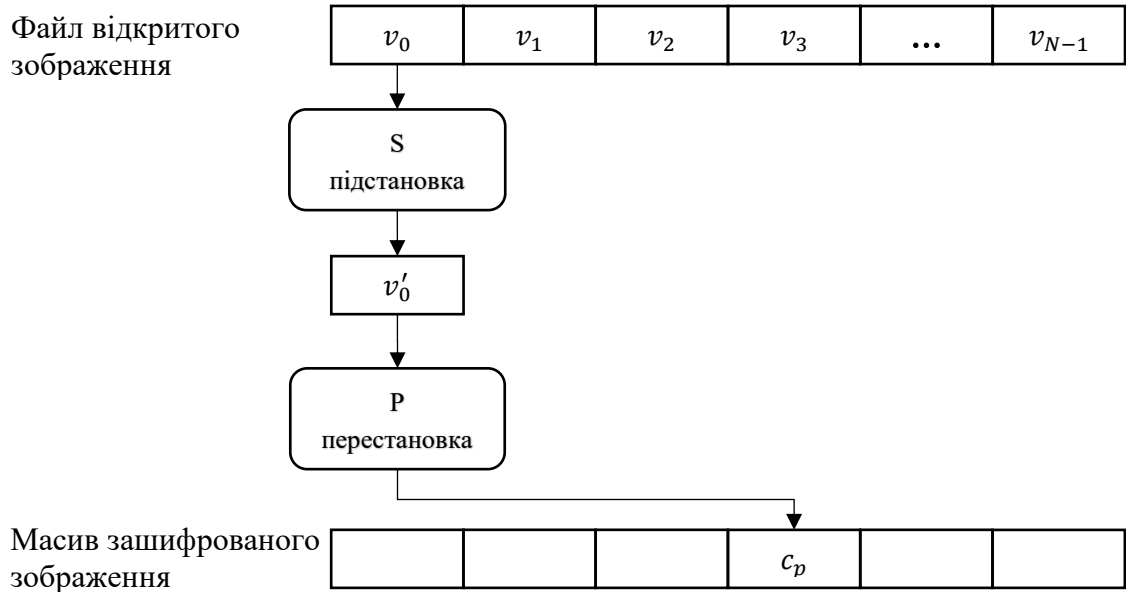


Рисунок 2.1 – Схема процесу зашифрування за методом SP

Зчитування з файлу можливе не лише по байтах, а блоками що складаються з кількох байтів. Оскільки, процес зашифрування передбачає зчитування з файлу лише по одному байту за один раз, то можна пришвидшити процес зашифрування за рахунок зчитування блоків байтів. Саме це, реалізує другий метод шифрування що пропонується.

Для зашифрування з файлу зчитуються блоки байтів, з яких формується масив відкритого зображення. Далі з цього масиву зчитуються байти в порядку, що відповідає правилу перестановки:

$$\pi_{заш} = \begin{pmatrix} p_0, p_1, \dots, p_{N-1} \\ 0, 1, \dots, N-1 \end{pmatrix}.$$

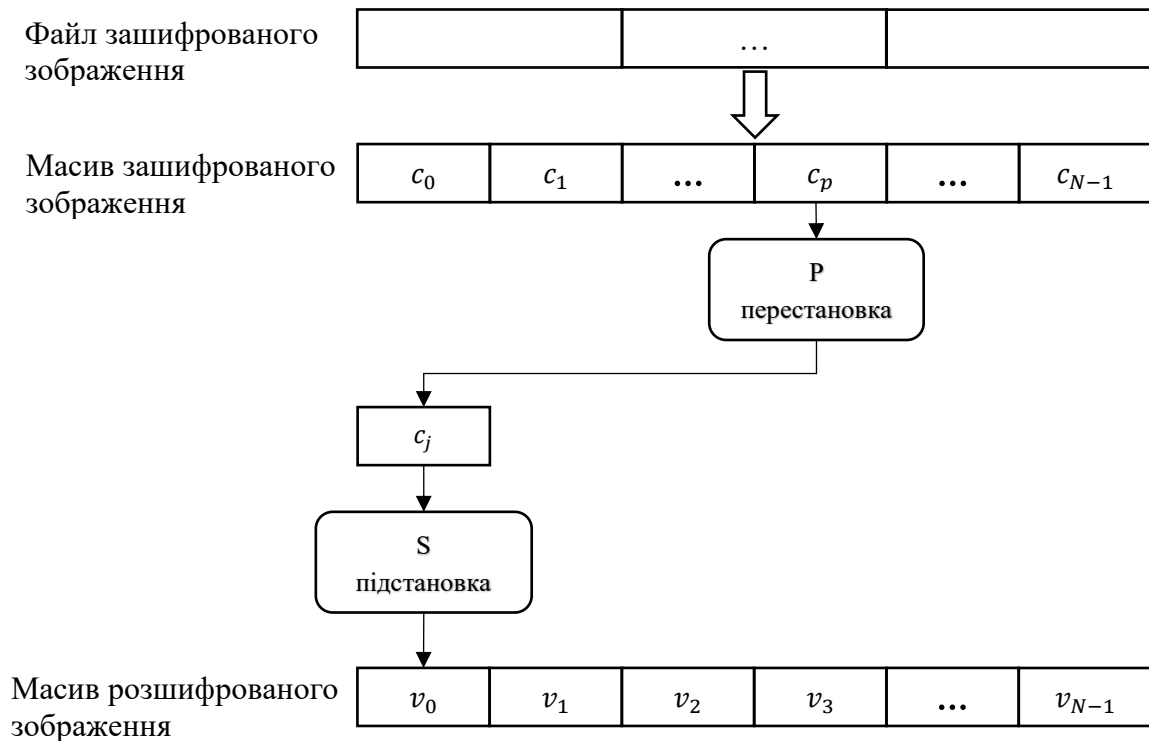


Рисунок 2.2 – Схема процесу розшифрування за методом SP

При цьому для кожного байту застосовується правило підстановки:

$$C = S_{K_2}^{PS}(V').$$

Оскільки спочатку виконується перестановка, а потім підстановка, то даний метод шифрування будемо називати методом *PS*. З урахуванням цього функція зашифрування має вигляд:

$$E_K^{PS}(V) = S_{K_2}^{PS}(P_{K_1, K_3}(V)) = C.$$

Схему процесу такого зашифрування наведено на рис. 2.3.

Для розшифрування з файлу зашифрованого зображення зчитуються блоки байтів, з яких формується масив зашифрованого зображення. Далі з цього масиву зчитуються байти в порядку, що відповідає оберненому правилу перестановки:

$$\pi_{розш} = \begin{pmatrix} 0, 1, \dots, N-1 \\ p_0, p_1, \dots, p_{N-1} \end{pmatrix}.$$

При цьому для кожного байту застосовується правило оберненої підстановки $V' = S_{K_2}^{-1}(C)$.

Виходячи з цього функція розшифрування має вигляд:

$$D_K^{PS}(C) = P_{K_1, K_3}^{-1}(S_{K_2}^{-1}(C)) = V.$$

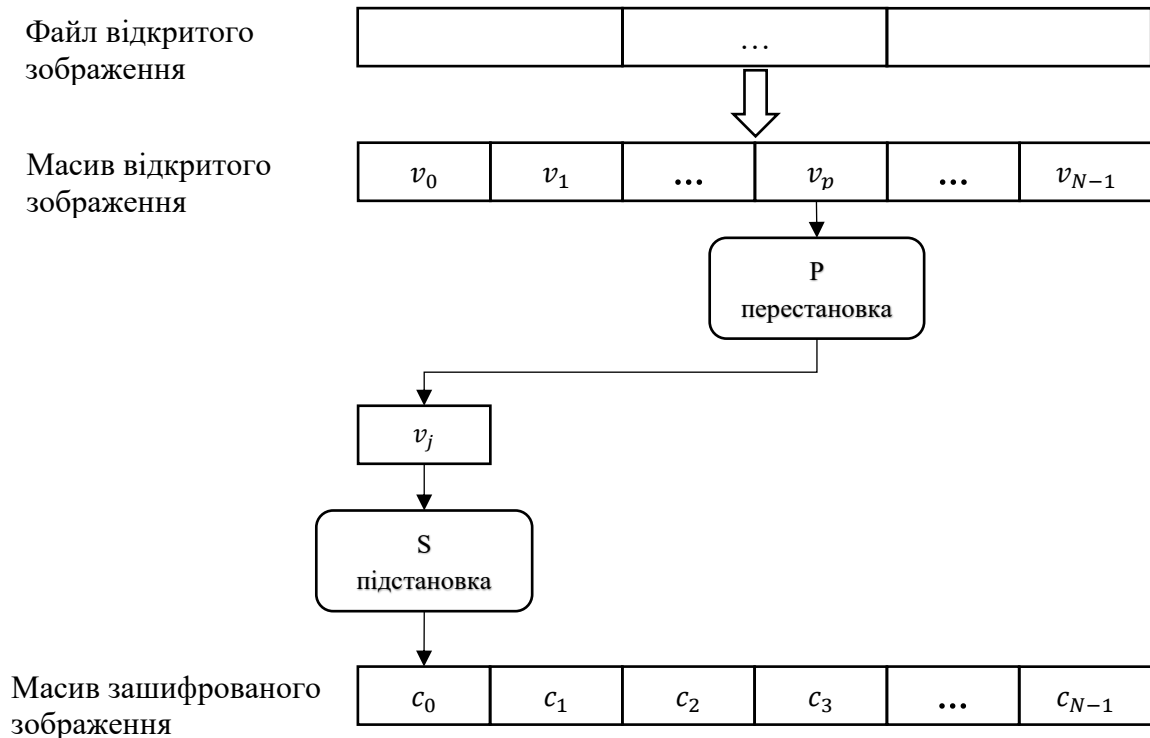


Рисунок 2.3 – Схема процесу зашифрування за методом *PS*

Схему процесу розшифрування за методом *PS* наведено на рис. 2.4.

Перевагою методу шифрування *PS* є пришвидшення процесу зашифрування, але це вимагає використовувати додатковий масив для відкритого зображення, що призводить до збільшення використовуваного обсягу пам'яті. Тому в тих випадках, коли є обмеження на обсяг використовуваної пам'яті, доцільно використовувати метод *SP*. Метод *PS* рекомендується використовувати, коли немає обмежень на обсяг використовуваної пам'яті, але є вимога до швидкості шифрування.

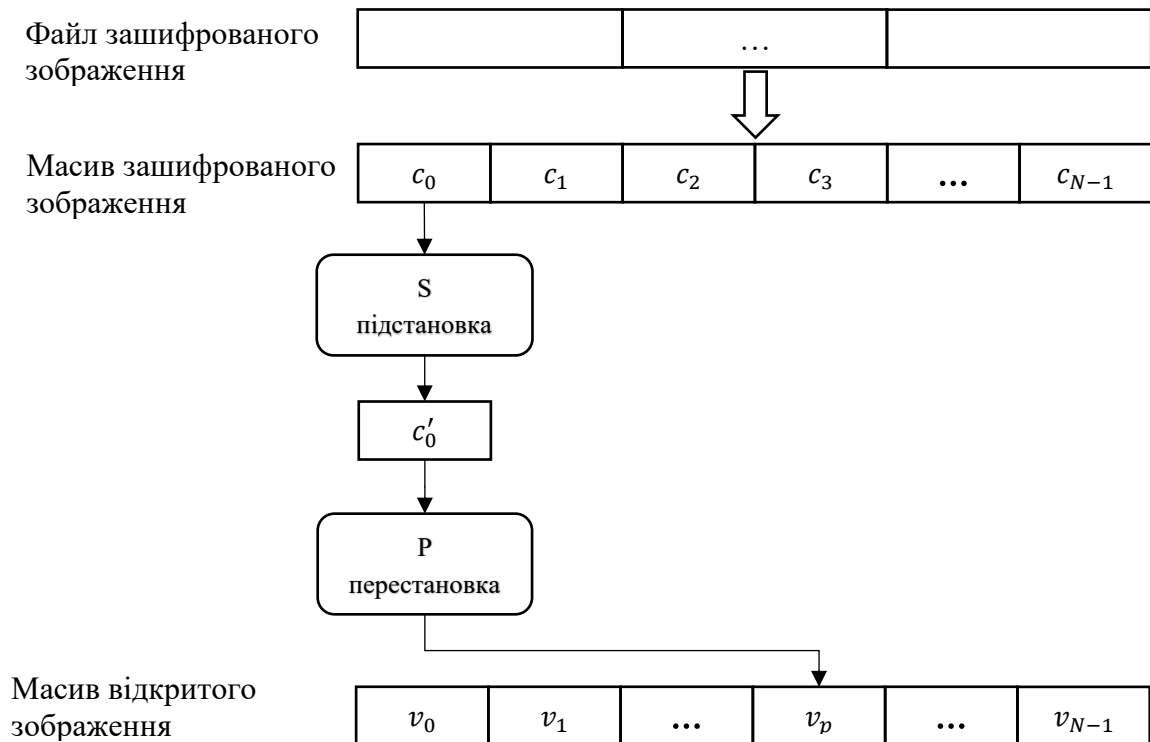


Рисунок 2.4 – Схема процесу розшифрування за методом *PS*

2.2.1 Реалізація підстановок

Для реалізації операції підстановки використовуються базові арифметичні операції над елементами поточного вектора зображення. Такий підхід дає змогу не збільшувати обчислювальні витрати за рахунок складних алгебраїчних перетворень, але водночас забезпечує зміну числових значень елементів відповідно до маскувальної послідовності, що формується на основі секретного ключа.

Залежно від обраного порядку виконання операцій підстановка може застосовуватися або безпосередньо до вхідного вектора V , або до проміжного вектора V' , отриманого після виконання перестановки. Тому для двох варіантів зашифрування використовуються відповідні позначення $S_{K_2}^{SP}$ та $S_{K_2}^{PS}$.

У варіанті SP для кожного елемента v_i формується змінене значення шляхом додавання маскувальної величини $Z(t)$ за модулем 256:

$$S_{K_2}^{SP}(\mathbf{V}) = (\mathbf{V}_i + Z(t)) \bmod 256 .$$

У варіанті PS, коли спочатку виконується перестановка, підстановка застосовується вже не до початкового вектору $\mathbf{V}$, а до проміжного вектора $\mathbf{V}'$, сформованого після перестановки. У цьому випадку правило має вигляд:

$$S_{K_2}^{PS}(\mathbf{V}') = (\mathbf{V}'_i + Z(t)) \bmod 256 .$$

Маскувальна величина $Z(t)$ формується на кожному кроці t на основі поточного стану регістра зсуву та параметрів, що визначаються секретним ключем:

$$Z(t) = (aX(t) + bY(t) + c) \bmod 256 .$$

тут a , b , c – непарні числа, задані відповідною складовою секретного ключа. Використання непарних коефіцієнтів доцільне з погляду збереження впливу компонентів $X(t)$ та $Y(t)$ на результат обчислення $Z(t)$ у просторі значень за модулем 256. Оскільки обчислення виконується над 8-бітовими значеннями, результат завжди належить діапазону:

$$Z(t) \in \{0, 1, \dots, 255\} .$$

Значення $X(t)$ та $Y(t)$ отримуються зі стану регістра $R(t)$. Для формування $X(t)$ використовуються старші 8 бітів стану регістра після відповідного зсуву:

$$X(t) = R(t) \& 0xFF .$$

Для формування $Y(t)$ використовується наступна 8-бітова частина стану, що отримується після додаткового зсуву:

$$Y(t) = (R(t) \gg 8) \& 0xFF .$$

Таким чином, на кожному кроці підстановки з поточного стану регістра формується дві 8-бітові величини $X(t)$ та $Y(t)$, які далі використовуються для обчислення маскувального значення $Z(t)$. Це значення є залежним від стану генератора та параметрів секретного ключа, тому при однаковому значенні ключа

та однаковій початковій ініціалізації формується однакова маскувальна послідовність, що забезпечує відтворюваність процесу підстановки під час розшифрування.

Оскільки значення елемента вектора та маскувальна величина належать одному діапазону 8-бітових значень, результат підстановки також залишається в межах допустимого діапазону:

$$(v_i + Z(t)) \bmod 256 \in \{0, 1, \dots, 255\}.$$

Тому підстановка не потребує додаткового нормування значень і може застосовуватися безпосередньо до компонентів зображення, представлених у вигляді байтів.

Обернена підстановка виконується над елементами проміжного вектора V' . Якщо під час прямої підстановки значення елемента змінювалося за правилом додавання маскувальної величини $Z(t)$ за модулем 256, то обернене перетворення виконується шляхом віднімання тієї самої величини:

$$v_i = (v'_i - Z(t)) \bmod 256.$$

2.2.2 Реалізація операції перестановки

Перестановка байтів поточного вектора у межах його повного обсягу передбачає формування впорядкованої послідовності індексів, яка визначає нові позиції елементів. Така послідовність має бути псевдовипадковою та не містити повторів, тобто відповідати перестановці множини індексів $\{0, 1, \dots, N-1\}$. Псевдовипадковість забезпечується використанням секретного ключа, який визначає параметри генерації індексів.

У роботі [87] запропоновано узагальнений підхід до побудови генераторів чисел у діапазоні від 0 до $N-1$. Основна ідея цього підходу полягає в тому, що діапазон чисел від 0 до $N-1$ розбивається на L піддіапазонів. Якщо $b = \frac{N}{L}$ – ціле число, то кожен з піддіапазонів складається з b чисел. Кожен піддіапазон

визначається мінімальним та максимальним значенням числа:

$$\mathbf{B}_l = [b_{\min}^l, b_{\max}^l]$$

Якщо b не є цілим числом, то $(L-1)$ піддіапазонів складаються з $\left\lceil \frac{N}{l} \right\rceil$ чисел, а L -й піддіапазон містить $N - l(l-1) \left\lceil \frac{N}{l} \right\rceil$ чисел.

У запропонованому підході використовується регістр зсуву з лінійним зворотним зв'язком (РЗЛЗЗ), який є генератором псевдовипадкової бітової послідовності. Регістр розглядається як детермінований автомат зі станом фіксованої довжини в 64 біти, який оновлюється на кожному кроці відповідно до заданої структури зворотного зв'язку.

Початковий стан регістра задається складовою секретного ключа K_1 , що забезпечує відтворюваність сформованої послідовності під час процесу розшифрування. На кожному кроці t зі стану регістра формується двійковий код фіксованої довжини r , який отримується шляхом вибору r бітів поточного стану (наприклад, старших бітів).

Цей r -розрядний код інтерпретується як номер одного з піддіапазонів. Оскільки кількість можливих значень такого коду дорівнює 2^r , кількість піддіапазонів визначається як:

$$L = 2^r.$$

Таким чином, РЗЛЗЗ не визначає структуру перестановки безпосередньо, а формує керуючу послідовність, що задає вибір піддіапазону на кожному кроці.

Вибір чисел з піддіапазону здійснюється детермінованим способом. При цьому можливі два варіанти. Перший полягає в формуванні послідовності чисел починаючи з $b_{\min}^l$ збільшенням числа на кожному кроці на одиницю. Другий варіант – це формування послідовності чисел починаючи з $b_{\max}^l$, зменшенням числа на одиницю на кожному кроці. Ці операції пропонується реалізувати на основі

реверсивного лічильника CT . Лічильник кожного піддіапазону формує числа від 0 до $b - 1$ або від $b - 1$ до 0. Наявність двох варіантів забезпечує додаткову стійкість до зламу, оскільки для кожного з лічильників можна окремо визначити початковий і кінцевий стани. Початкове значення стану лічильника визначається складовою секретного ключа $K_2 = \{k_{2,l}\}$, де $l = 0, 1, \dots, L - 1$. В табл. 2.1 наведено правило визначення початкових станів лічильників для ключа $K_2 = \{0, 1, 1, 0\}$.

Таблиця 2.1 – Правило визначення початкового стану CT

Секретний ключ $k_{2,l}$	Початковий стан $CT_l(0)$
0	$CT_0 := b_{\min}^0$
1	$CT_1 := b_{\max}^1$
1	$CT_2 := b_{\max}^2$
0	$CT_3 := b_{\max}^3$

Таблиця початкових станів лічильників показує, як біти складової ключа K_2 визначають напрям обходу кожного піддіапазону. Для піддіапазону B_l відповідний біт $k_{2,l}$ задає початкове значення лічильника CT_l : якщо $k_{2,l} = 0$, обхід починається з першої локальної позиції піддіапазону; якщо $k_{2,l} = 1$, обхід починається з останньої локальної позиції. Отже, K_2 не змінює самі межі піддіапазонів, а визначає порядок вибору індексів усередині кожного з них.

Відповідність між r -розрядним кодом, сформованим станом РЗЛЗЗ та поточним (обраним) піддіапазоном на кроці t для випадку $r=2$ наведено в табл. 2.2.

Таблиця 2.2 – Правило вибору піддіапазону (лічильника)

r -розрядний код РЗЛЗЗ	Номер піддіапазону	Поточний лічильник
00	$l = 0$	$CT_0(t)$
01	$l = 1$	$CT_1(t)$
10	$l = 2$	$CT_2(t)$
11	$l = 3$	$CT_3(t)$

Отриманий код не є остаточним індексом перестановки, а лише визначає піддіапазон, з якого цей індекс буде сформовано за допомогою відповідного

лічильника.

Зміна стану лічильника визначається за таким правилом:

$$CT_l(t+1) = CT_l(t) + (-1)^{K_2, l},$$

де $CT_l(t)$ – стан l -го лічильника на кроці t .

Правило зміни стану лічильників для випадку $r = 2$ і $K_2 = \{0, 1, 1, 0\}$, наведено в табл. 2.3.

Таблиця 2.3 – Правило зміни стану лічильників CT

K_2	Зміна стану
0	$CT_0 := CT_0 + 1$
1	$CT_1 := CT_1 - 1$
1	$CT_2 := CT_2 - 1$
0	$CT_3 := CT_3 + 1$

Таке правило узгоджує початковий стан лічильника з напрямом обходу піддіапазону і запобігає повторному вибору вже використаної позиції в межах B_l .

Число, що обирається з l -го піддіапазону визначається виходячи з номера піддіапазону і стану відповідного лічильника:

$$p = l \cdot b + CT_l(t).$$

На рис. 2.5 зображена загальна схеми виконання операції перестановки на кроці t .

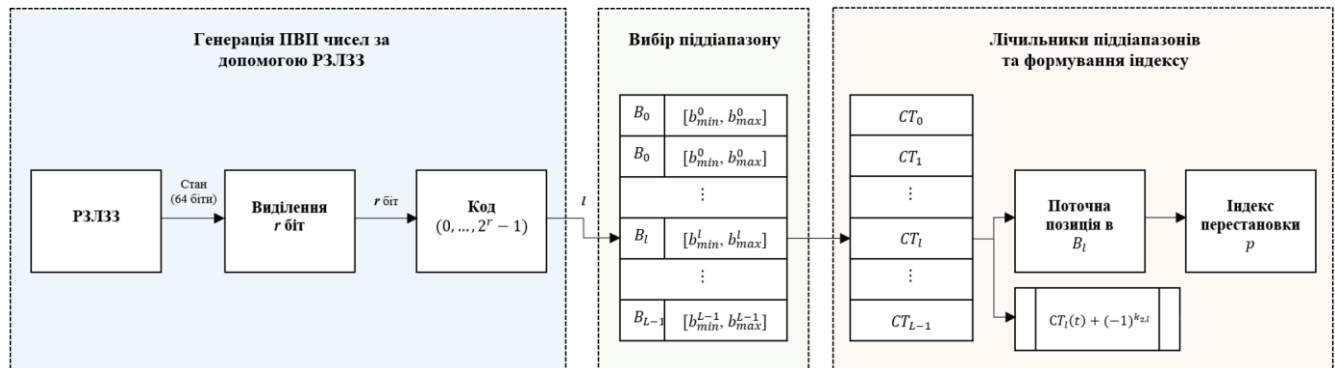


Рисунок 2.5 – Схеми процедури перестановки

При використанні лічильників разом з генератором ПВП, способом який описано вище, може виникати колізія, коли в поточному піддіапазоні вектора C всі

числа вже були обрані на попередніх кроках та генератор вказує на даний піддіапазон знову. Така ситуація може спотворити процес зашифрування/розшифрування вектору шляхом перезапису вже раніше обраного числа, що призведе до неможливості відновлення вхідного зображення. Для усунення цього пропонується додатковий механізм перевірки поточного стану лічильника. Якщо $CT_l(t) = b - 1$ при $K_2, l = 0$ або $CT_l(t) = 0$ при $K_2, l = 1$, то формування відповідного числа з піддіапазону l не відбувається. Процес формування псевдовипадкової послідовності чисел завершується, коли вказані вище умови виконуються для усіх лічильників.

2.2.3 Структура секретного ключа

Секретний ключ **K** має три складові:

$$\mathbf{K} = \{K_1, K_2, K_3\},$$

які визначають параметри, необхідні для реалізації підстановок та перестановок. Ключ використовується як набір керуючих величин, що визначають послідовність вибору піддіапазонів, порядок перебору індексів у межах піддіапазонів, а також параметри формування маскувальної послідовності підстановки.

Компонент K_1 визначає початковий стан генератора псевдовипадкової послідовності на основі РЗЛЗЗ. Цей стан використовується для формування керуючої послідовності, яка на кожному кроці задає номер піддіапазону l , з якого виконується вибір наступного індексу перестановки. Таким чином, K_1 фіксує детерміновану послідовність вибору піддіапазонів.

Компонент K_2 визначає:

- початкові значення лічильників $CT_l(0)$ для піддіапазонів;
- режим роботи лічильників, тобто напрямок оновлення:

$$\sigma_l \in \{+1, -1\},$$

що відповідає додаванню або відніманню одиниці при переході до наступної позиції.

Ці параметри визначають порядок перебору індексів у межах кожного піддіапазону та є необхідними для формування перестановки без повторів і з повним покриттям індексного простору.

Компонент K_3 задає значення a , b , c , які використовуються для обчислення маскувальної величини $Z(t)$ для виконання процесу підстановки.

Отже, набір параметрів K_1 , K_2 , K_3 задає повний набір даних, достатній для відтворення перетворень зашифрування та розшифрування.

2.3 Методи розподілу секретного вмісту зображення

Після виконання процедури шифрування вхідне зображення подається у вигляді зашифрованого вектора даних, який може розглядатися як секрет, підготовлений до подальшого розподілу між учасниками схеми. Подальша обробка не впливає на криптографічні властивості зашифрованих даних, а лише визначає спосіб їх зберігання та відновлення.

Такий підхід дозволяє чітко розмежувати функції окремих етапів: шифрування відповідає за забезпечення криптографічної стійкості, тоді як схема розподілу визначає модель доступу до даних, умови відновлення та характеристики часток, що передаються учасникам.

У межах даної роботи розглядаються два підходи до розподілу зашифрованого зображення, які базуються на різних моделях доступу, але застосовуються до одного й того самого результату попереднього шифрування. Перший підхід ґрунтується на базовій схемі розподілу типу (n,n) , у якій кожна частка містить фрагмент зашифрованих даних, а відновлення можливе лише за наявності всіх часток. Другий підхід реалізує 2 варіанти порогової (k,n) -схеми, у якій відновлення шифротексту здійснюється за участі не менше ніж k учасників.

2.3.1 Метод розподілу секретного вмісту зображення за (n,n) -схемою

Після зашифрування вхідного зображення вектор $\mathbf{C} = \{c_0, c_1, \dots, c_{N-1}\}$

розбивається на n підвекторів із метою організації розподіленого зберігання без можливості відновлення повного зображення за наявності лише частини підвекторів. Для цього обрано схему розподілу секрету типу (n,n) , яка гарантує, що лише за наявності усіх n частин можливе коректне відновлення вихідного вмісту, що підвищує рівень інформаційної безпеки.

Для розподілення на підвектори пропонується використовувати циклічний рівномірний розподіл, який забезпечує однорідне заповнення даними кожного підвектору та дозволяє зберегти лінійну відповідність між елементами оригінального та зашифрованого вектора, що спрощує процес розшифрування та мінімізує накладні обчислення.

Кожен підвектор розподіленого зображення:

$$P_y = \{p_{y,0}, p_{y,1}, \dots, p_{y,q}\},$$

утворюється з елементів вектора C :

$$p_{y,q} = c_i, \quad \text{де } y = i \bmod n, \quad q = y + \left\lfloor \frac{i}{n} \right\rfloor \cdot n.$$

Таким чином, утворюється послідовність підвекторів $\mathbf{P} = \{P_0, P_1, \dots, P_{n-1}\}$,

розміри яких не перевищують $\frac{N}{n} \pm 1$.

Для відновлення вхідного зображення обов'язковою є наявність повного набору часток $\{P_0, P_1, \dots, P_{n-1}\}$.

Оскільки для розподілу використовувався циклічний рівномірний підхід, індекси елементів зашифрованого вектора було призначено у детермінованому порядку, що зберігає взаємно однозначну відповідність між елементами кожної частини P_y та їх позиціями в C . Це дозволяє сформулювати зворотне перетворення, яке відновлює C шляхом простого об'єднання фрагментів за фіксованими індексами відповідно до початкового порядку розподілу.

Для кожної частини $P_y = \mathbf{C}$, відновлення елемента вектора $\mathbf{C}$ відбувається за формулою:

$$c_i = p_{y,q}, \quad \text{де } y = i \bmod n, \quad q = y + \left\lfloor \frac{i}{n} \right\rfloor \cdot n.$$

Побудована схема розподілу за моделлю (n,n) у межах запропонованого підходу виконує функцію рівномірного рознесення зашифрованого вмісту між учасниками без внесення додаткової надлишковості та без ускладнення базової логіки розподілу. Такий спосіб дозволяє зберегти детермінованість процесу відновлення та мінімізувати службові дані, оскільки порядок формування часток однозначно визначається індексним правилом розподілу.

Використання (n,n) -схеми у даному випадку не спрямоване на досягнення порогових властивостей доступу, а розглядається як базовий механізм розподілу зашифрованого зображення між усіма учасниками. Основна криптографічна стійкість забезпечується попереднім етапом шифрування, тоді як схема (n,n) гарантує коректне та повне відновлення вихідного зображення лише за наявності всіх сформованих часток.

2.3.2 Метод розподілу секретного вмісту зображення за (k,n) -схемою з використанням коаліцій учасників

Метод розподілу секретного вмісту зображення за пороговою (k,n) -схемою передбачає можливість відновлення за наявності не менш ніж k валідних часток із загальної кількості n . Побудова схем розподілу секрету для заданих структур доступу належить до класичних напрямів теорії розподілу секрету. У роботах [88, 89] було показано можливість реалізації узагальнених структур доступу, де право відновлення визначається не лише значенням порогу, а множинами допустимих коаліцій учасників. Подальший розвиток таких конструкцій пов'язаний із кумулятивними схемами та комбінаторними способами призначення частин секрету учасникам.

Автором пропонується узагальнений метод розподілу секретного вмісту зображення за схемою (k, n) , у якому структура часток формується відповідно до визначеної множини коаліції, а відновлення зашифрованого вектору виконується за наявності не менше ніж k часток і використання додаткової складової у вигляді частки дилера.

Метою розподілу є формування множини часток $\mathbf{P}$ учасників $\mathbf{U} = \{U_1, U_2, \dots, U_n\}$. Для методу приймається така залежність параметрів k та n :

$$2 \leq k < n.$$

При цьому зашифрований вектор $\mathbf{C}$ обсягом O_C байт поділяється на $m = \binom{n}{k} = \frac{n!}{k!(n-k)!}$ блоків обсягом $\frac{O_C}{m}$ кожен. Кількість блоків відповідає кількості всіх можливих коаліцій із k учасників, які можуть бути утворені з множини n учасників. Для визначення правил розміщення блоків між частками учасників спочатку задається множина індексів учасників $\mathbf{I} = \{1, 2, \dots, n\}$. На основі цієї множини формується впорядкований набір індексних множин $\mathbf{G} = \{G_1, G_2, \dots, G_m\}$. Кожна множина G_j містить k різних індексів учасників і відповідає одній можливій коаліції потужності k . Побудова набору $\mathbf{G}$ виконується послідовним генеруванням усіх комбінацій із k індексів множини $\mathbf{I}$. Початковою комбінацією є $G_1 = \{1, 2, \dots, k\}$.

Нехай поточна комбінація подана впорядкованим набором $\mathbf{g} = \{g_1, g_2, \dots, g_k\}$, де $g_1 < g_2 < \dots < g_k$. Для отримання наступної комбінації індекси переглядаються справа наліво. Знаходиться найбільший номер позиції t , для якого поточне значення ще може бути збільшене без порушення порядку індексів: $g_t < n - k + t$. Після цього значення g_t збільшується на одиницю, а всі елементи праворуч від нього встановлюються мінімально можливими послідовними значеннями: $g_t := g_t + 1$, $g_{t+\mu} := g_t + \mu$, $\mu = 1, 2, \dots, k - t$. Отриманий набір індексів утворює наступну

множину G_j . Процес повторюється до отримання останньої комбінації:

$$G_m = \{n - k + 1, n - k + 2, \dots, n\}.$$

Такий порядок формування $\mathbf{G}$ є детермінованим і може бути відтворений під час відновлення за відомими параметрами k та n . Після формування набору $\mathbf{G}$ кожному блоку зашифрованого вектора ставиться у відповідність одна індексна множина. Якщо $\mathbf{C} = \{C_1, C_2, \dots, C_m\}$, то блоку C_j відповідає множина G_j , де

$j=1, 2, \dots, m$. Множина G_j визначає, до яких часток блок C_j не включається. Якщо індекс учасника належить G_j , то блок C_j не записується до його частки. Якщо індекс учасника не належить G_j , то блок C_j включається до відповідної частки. Тому частка учасника U_q формується за правилом:

$$P_q = \{C_j \mid q \notin G_j\}.$$

Таким чином, кожен блок C_j не входить до часток рівно k учасників, індекси яких утворюють відповідну множину G_j , і входить до часток решти $n-k$ учасників.

Частка дилера D не входить до складу жодної з часток учасників і визначається як сума за модулем 2 усіх блоків C_j :

$$D = C_1 \oplus C_2 \oplus \dots \oplus C_m.$$

Відновлення зашифрованого вектора виконується за множиною часток учасників, що утворюють коаліцію:

$$\mathbf{A} \in \mathbf{U}, \quad |\mathbf{A}| = k.$$

Для цієї коаліції визначається множина індексів учасників $\mathbf{W}(\mathbf{A}) = \{q \mid U_q \in \mathbf{A}\}$.

Оскільки множина $\mathbf{G}$ містить усі комбінації індексів учасників з k елементів, для кожної коаліції $\mathbf{A}$ існує єдина індексна множина G_s , для якої виконується $G_s = \mathbf{W}(\mathbf{A})$.

Відповідно, існує єдиний блок C_s пов'язаний із цією множиною. За правилом формування часток блок C_s не входить до жодної з часток учасників коаліції $\mathbf{A}$, оскільки індекси всіх цих учасників належать множині G_s . Усі інші блоки входять щонайменше до однієї з часток учасників цієї коаліції, тому після об'єднання часток

отримується набір:

$$\mathbf{C}^A = \bigcup_{U_q \in A} P_q.$$

Для коаліції з k учасників цей набір містить усі блоки зашифрованого вектора, крім одного: $\mathbf{C}^A = \{C_j \mid j = 1, 2, \dots, m, j \neq s\}$. Отже, відсутнім є блок C_s , індекс якого визначається з умови $G_s = \mathbf{W}(A)$. Його відновлення виконується з використанням

частки дилера: $C_s = D \oplus \bigoplus_{\substack{j=1, \\ j \neq s}}^m C_j$. Після обчислення C_s формується повний

упорядкований набір блоків $\mathbf{C} = \{C_1, C_2, \dots, C_m\}$.

Розглянемо приклад застосування такого методу розподілу для схеми $k = 2$, $n = 4$.

Згідно з правилом розбиття, кількість блоків m дорівнює 6: $\mathbf{C} = C_1, C_2, C_3, C_4, C_5, C_6$. Далі необхідно сформулювати правило вибору блоків C_j ($j=1, 2, \dots, 6$) до часток P_i ($i=1, 2, 3, 4$) на основі визначення допустимих множин індексів:

$$G_1 = \{1, 2\}, G_2 = \{1, 3\}, G_3 = \{1, 4\}, G_4 = \{2, 3\}, G_5 = \{2, 4\}, G_6 = \{3, 4\}.$$

Як бачимо з прикладу, множини індексів $\mathbf{G}$ забезпечують повне покриття усіх блоків $\mathbf{C}$ таким чином, що кожен набір індексів зустрічається тільки один раз. На основі значень цих множин виконується формування часток учасників P :

$$P_1 = \{C_4, C_5, C_6\}, P_2 = \{C_2, C_3, C_6\}, P_3 = \{C_1, C_3, C_5\}, P_4 = \{C_1, C_2, C_4\}.$$

Частки P були сформовані таким чином, що в кожен частку входять тільки ті блоки $\mathbf{C}$ індекс яких відсутній у множині G_j .

Аналізуючи перетин будь-яких k часток $\mathbf{P}$ бачимо, що вони спроможні сформулювати таку послідовність блоків C_j , при якому завжди буде не вистачати одного блоку. Коаліція $A = \{U_1, U_3\}$ акумулює частки $P_1 = \{C_4, C_5, C_6\}$ та $P_3 = \{C_1, C_3, C_5\}$, при яких формується $\mathbf{C} = C_1, C_3, C_4, C_5, C_6$. В такому випадку

відсутнім блоком є C_2 , значення якого ми можемо відновити з використання частки дилера D :

$$C_2 = D \oplus C_1 \oplus C_3 \oplus C_4 \oplus C_5 \oplus C_6.$$

На даному етапі визначена коаліція учасників за участю дилера вже може відновити зашифрований вектор C . При цьому, якщо розглянути випадок коли коаліція налічує $k-1$ учасників, то відновлення відсутнього блоку з використання частки дилера є неможливим, оскільки вона не може відновити 2 чи більше невідомих складових.

2.3.3 Спрощений метод розподілу секретного вмісту зображення за пороговою схемою (k,n) з перекриттям блоків

Описаний вище метод розподілу секретного вмісту зображення за пороговою схемою (k,n) передбачає розбиття зашифрованого вектора на кількість блоків, що суттєво перевищує кількість учасників схеми розподілу. Крім того, процедура формування часток учасників є достатньо складною. Для усунення цих недоліків пропонується спрощений метод розподілу за схемою (k,n) , який базується на використанні простих операцій, таких як циклічний зсув для формування часток учасників.

Аналогічно до попереднього методу отримані дані представляються як набір блоків, але при цьому кількість яких дорівнює кількості учасників схеми розподілу:

$$C = \{c_1, c_2, \dots, c_n\}.$$

Якщо довжина зашифрованого вектора не ділиться на n націло, останній блок може бути доповнений до довжини інших блоків. Для формування часток учасників вводиться величина:

$$p = n - k,$$

яка визначає кількість блоків, що включаються до однієї індивідуальної частки. Тоді для кожного учасника формується частка:

$$P_q = \{c_{\alpha_0}, c_{\alpha_q}, \dots, c_{\alpha_{p-1}}\},$$

де: $q = 1, 2, \dots, n$;

$$\alpha_\tau = ((q + \tau - 1) \bmod n) + 1;$$

$$\tau = 1, 2, \dots, p - 1.$$

Таким чином, частка P_q містить p послідовних блоків зашифрованого вектора, починаючи з блоку c_q , а при досягненні кінця послідовності використовуються блоки починаючи з першого.

Дилер додатково формує значення:

$$D = c_1 \oplus c_2 \oplus \dots \oplus c_n.$$

Складова D не входить до складу жодної індивідуальної частки P_q і не передається учасникам на етапі розподілу, а зберігається окремо у дилера. Вона використовується лише під час санкціонованого відновлення.

Наприклад, якщо маємо порогову схему $(k, n) = (3, 5)$, то $p = 5 - 3 = 2$ і частки учасників складаються з двох елементів. Оскільки, вектор C розбивається на 5 блоків $C = \{c_1, c_2, \dots, c_5\}$, то маємо такі частки учасників:

$$P_1 = \{c_{1+j}\} = \{c_1, c_2\},$$

$$P_2 = \{c_{2+j}\} = \{c_2, c_3\},$$

$$P_3 = \{c_{3+j}\} = \{c_3, c_4\},$$

$$P_4 = \{c_{4+j}\} = \{c_4, c_5\},$$

$$P_5 = \{c_{5+j}\} = \{c_5, c_1\}.$$

Така структура забезпечує перекриття між частками, однак повне відновлення зашифрованого вектора в запропонованій схемі виконується не лише за рахунок цих перекриттів, а з використанням допоміжної частки дилера.

Відновлення зашифрованого вектора виконується після отримання часток від коаліції учасників, для якої виконується порогова умова $|e| \geq k$, при

$$A = \{P_{q_1}, P_{q_2}, \dots, P_{q_e}\}.$$

Процес відновлення як і розподілу відбувається на стороні дилера, який агрегує усі надані йому частки, і аналізує сукупність блоків, фактично отриманих із цих часток.

Залежно від того, які блоки доступні після об'єднання часток допустимої коаліції, можливі два основні режими відновлення.

Перший режим відповідає випадку, коли всі блоки зашифрованого вектора вже наявні серед отриманих часток. У цьому разі відновлення виконується без використання дилерської складової. Блоки впорядковуються за їх номерами та послідовно об'єднуються у вектор C . Для схеми (3,5) розглянемо набір часток:

$$P_1\{c_1, c_2\}, P_3\{c_3, c_4\}, P_5\{c_5, c_1\}.$$

Після їх об'єднання отримується повний набір блоків:

$$P_1 \cup P_3 \cup P_5 = \{c_1, c_2, c_3, c_4, c_5\}.$$

Другий режим виникає тоді, коли після об'єднання часток допустимої коаліції відсутній рівно один блок. У цьому випадку повне збирання вектора неможливе, тому використовується дилерська складова. Оскільки, дилерська частка містить у собі суму за модулем 2 значень усіх інших блоків, то відсутній блок визначається як:

$$C_m = D \oplus \bigoplus_{i \neq m} C_i. \quad (2.1)$$

Після цього виконується послідовне збирання вектора.

Для тієї самої схеми (3,5) візьмемо частки $P_1\{c_1, c_2\}$, $P_2\{c_2, c_3\}$, $P_3\{c_3, c_4\}$, в даному випадку їх об'єднання дає нам: $P_1 \cup P_2 \cup P_3 = \{c_1, c_2, c_3, c_4\}$. Для відновлення вектору C не вистачає блоку c_5 .

Використавши вираз (2.1), можемо також відновити значення блоку c_5 :

$$c_5 = D \oplus c_1 \oplus c_2 \oplus c_3 \oplus c_4$$

Множина з менш ніж k часток не є допустимою для відновлення. Якщо для відновлення дилеру надано $k - 1$ часток, то сумарна кількість блоків буде

дорівнювати $n - 2$. На основі власної частки дилер може визначити лише один блок, а в даному випадку потрібно визначити 2 блоки. Тобто дилер не має можливості відновити C за наявності $k - 1$ часток.

Таким чином, процес відновлення зашифрованого вектора має два режими. У першому режимі всі блоки отримуються безпосередньо з часток допустимої коаліції, після чого виконується їх конкатенація. У другому режимі з часток отримуються всі блоки, крім одного, а відсутній блок обчислюється з використанням дилерської складової. Перекриття часток забезпечує надлишковість на рівні блоків, а дилерська складова завершує відновлення тоді, коли цієї надлишковості недостатньо для повного відновлення.

Така реалізація схеми розподілу на відміну від описаного узагальненого методу можлива лише для певних значень параметрів k та n . Для коректного застосування необхідно використовувати допустимі конфігурації, за яких жодна множина з менш ніж k часток без використання D не забезпечує повного покриття всіх блоків C . Достатньою умовою для цього є:

$$(k - 1)(n - k) < n \quad (2.2)$$

Ця умова передбачає, що навіть за максимально сприятливого для несанкціонованого відновлення вибору $k - 1$ часток сумарна кількість блоків, які можуть бути ними покриті, є меншою за кількість усіх блоків зашифрованого вектора. Наприклад, для параметрів (3, 5):

$$(k - 1)(n - k) = 2 \cdot 2 = 4 < 5,$$

тому дві частки без доступу до D не можуть містити повний набір блоків C . Натомість для параметрів (3, 6):

$$(k - 1)(n - k) = 2 \cdot 3 = 6,$$

що не задовольняє умову (2.2) і тому реалізувати (3,6)-схему розподілу неможливо.

Приклади параметрів k та n , які задовольняють описану вище вимогу підбору: (2,3), (2,4), (2,5) і т.д. з будь яким значенням n ; (3,4), (3,5), (4,5), (4,6), (5,6),

(6,7).

2.4 Властивості запропонованого методу

2.4.1 Детермінованість, оборотність і коректність відновлення

Запропонований метод є детермінованим, оскільки всі етапи обробки зображення задаються вхідними даними, секретним ключем і параметрами відповідної моделі розподілу. За однакового зображення, однакового ключа та однакових параметрів схеми формується той самий зашифрований вектор і той самий набір часток. Ця властивість є необхідною умовою коректного відновлення, оскільки під час розшифрування мають бути відтворені ті самі послідовності підстановки, вибору піддіапазонів, станів лічильників і позицій перестановки, які використовувалися під час зашифрування.

Нехай F – файл вхідне зображення, V – його векторне подання, K – секретний ключ, а Γ – обрана модель розподілу. Модель Γ визначає спосіб формування часток із зашифрованого вектора: (n,n) -схему або (k,n) -схему за участю дилера. Для (k,n) -схеми за участю дилера додатково формується допоміжна складова D , яка не входить до складу індивідуальних часток.

Детермінованість прямого перетворення означає, що для фіксованих F , K та Γ виконується:

$$F_{K,\Gamma}(I) = \mathfrak{R}_{\Gamma},$$

де $\mathfrak{R}_{\Gamma}$ – множина часток, сформованих відповідно до вибраної моделі.

Якщо ж змінюється хоча б один із параметрів ключа, порядок виконання операцій, початковий стан РЗЛЗЗ або параметри моделі розподілу, то змінюється зашифрований вектор або структура сформованих часток.

Оборотність методу забезпечується наявністю оборотних перетворень для кожного етапу обробки.

Нехай після відновлення часток отримано зашифрований вектор:

$$\mathbf{C} = E_K(\mathbf{V}).$$

Тоді за наявності правильного ключа K і відтворення параметрів зашифрування існує обернене перетворення E_K^{-1} , для якого виконується $E_K^{-1}(\mathbf{C}) = \mathbf{V}$. Оскільки, $\mathbf{C} = E_K(\mathbf{V})$ то маємо $E_K^{-1}(E_K(\mathbf{V})) = \mathbf{V}$.

Це означає, що після відновлення зашифрованого вектора розшифрування відновлює саме те векторне подання, яке було отримане на початковому етапі перед зашифруванням.

Коректність відновлення залежить від обраної моделі розподілу. Для (n,n) -схеми повний зашифрований вектор відновлюється за наявності всіх сформованих часток:

$$R_{n,n}(\mathfrak{R}_{n,n}) = \mathbf{C}.$$

Для (k,n) - схеми за участю дилера відновлення виконується за наявності допустимої множини валідних часток і допоміжної дилерської складової.

$$\text{Якщо } \mathbf{A} \subseteq \mathfrak{R}_{k,n} \text{ і } |\mathbf{A}| \geq k, \text{ то } R_{k,n}(\mathbf{A}, D) = \mathbf{C}.$$

Отже, коректність методу визначається послідовним виконанням двох умов: коректною відновлення зашифрованого вектора $\mathbf{C}$, правильним застосуванням оберненого криптографічного перетворення E_K^{-1} . У загальному вигляді для (n,n) - схеми це можна подати як:

$$E_K^{-1}(R_{n,n}(\mathfrak{R}_{n,n})) = \mathbf{F},$$

а для (k,n) - схеми за участю дилера – як:

$$E_K^{-1}(R_{k,n}(\mathbf{A}, D)) = \mathbf{F}.$$

2.4.2 Відсутність доступу до відкритого зображення за окремими частками

Захисна властивість запропонованого методу визначається не лише способом розподілу даних між учасниками, а передусім порядком виконання перетворень. Оскільки розподіл виконується не над початковим зображенням і не над його

відкритим векторним поданням V , а над зашифрованим вектором C , то індивідуальні частки містять фрагменти або структурні компоненти вже у зашифрованому відображенні, а не безпосередні значення байтів початкового зображення. Це принципово відрізняє запропонований підхід від схем, у яких секрет розподіляється без попереднього криптографічного перетворення. Тобто, послідовність формування часток має вигляд $F \rightarrow V \rightarrow C \rightarrow \mathfrak{R}_\Gamma$.

У випадку (n,n) -схеми окрема частка містить лише частину елементів зашифрованого вектора. За відсутності інших часток неможливо відновити вектор C , а отже неможливо виконати обернене перетворення $V = E_K^{-1}(C)$.

Навіть якщо окрема частка містить певну підмножину байтів зашифрованого вектора, то ці байти не відповідають безпосередньо початковим значенням байтів, оскільки перед розподілом були застосовані підстановка та перестановка.

Для (k,n) -схеми за участю дилера аналогічна властивість визначається двома чинниками. По-перше, індивідуальні частки формуються з уже зашифрованих блоків C . По-друге, допоміжна дилерська складова D не входить до складу індивідуальних часток і не передається учасникам на етапі розподілу. Тому множина з менш ніж k часток не забезпечує відновлення без доступу до D . Така множина не може завершити відновлення зашифрованого вектора C , а без C неможливо отримати початковий вектор V .

Отже, захист у запропонованому методі забезпечується композицією двох етапів: спочатку початкове зображення перетворюється у зашифрований вектор C , після чого вже цей вектор розподіляється між учасниками. Наслідком цього є те, що частки не містять безпосереднього подання відкритого зображення, а відновлення F можлива лише після відновлення вектора C , застосування оберненого криптографічного перетворення зі знанням секретного ключа.

2.5 Оцінка обсягу часток у запропонованих (k,n) -схемах

Однією з характеристик методу розподілення секрету є обсяг часток, що надсилаються учасникам та зберігаються ними.

В узагальненому методі зашифрований вектор поділяється на кількість блоків, що відповідає кількості всіх коаліцій учасників:

$$m_3 = \binom{n}{k}.$$

Кожен блок пов'язується з однією коаліцією потужності k . Обсяг індивідуальної частки учасника становить:

$$O_q^3 = \frac{n-k}{n} O_C,$$

де O_C – обсяг зашифрованого вектора C .

Оскільки $k < n$, то $\frac{n-k}{n} < 1$. Отже, кожна окрема частка має менший обсяг, ніж обсяг зашифрований вектор C . Це підтверджує те, що кожен окремий учасник не зберігає повну копію зашифрованого зображення.

Сумарний обсяг індивідуальних часток у коаліційному методі дорівнює:

$$O_\Sigma^3 = (n-k) O_C.$$

Оскільки частка дилера має обсяг одного блоку зашифрованого вектору, то повний обсяг з урахуванням часток учасників та дилера становить:

$$O_\Sigma^3 = (n-k) O_C + O_q^3.$$

У спрощеному методі з циклічним перекриттям зашифрований вектор поділяється на n блоків. Кожна індивідуальна частка містить $n-k$ блоків, вибраних за циклічним правилом. Тому обсяг однієї частки визначається як:

$$O_q^c = \frac{(n-k)}{n} O_C.$$

Отже, як і в узагальненому методі, індивідуальна частка у спрощеній схемі є меншою за повний зашифрований вектор. Сумарний обсяг усіх індивідуальних часток дорівнює $O_{\Sigma}^c = (n-k)O_C$, дилерська частка у даному методі також займає обсяг у n блоків, тому повний обсяг розподілених даних становить:

$$O_{\Sigma}^c = (n-k)O_C + \frac{O_C}{n}.$$

Таким чином, узагальнений та спрощений методи мають однаковий обсяг індивідуальної частки:

$$O_q^3 = O_q^c = \frac{(n-k)}{n}O_C,$$

а також однаковий сумарний обсяг індивідуальних часток $O_{\Sigma}^3 = O_{\Sigma}^c = (n-k)O_C$.

Відомі методи розподілу за (k,n) -схемою використовують частки, обсяг кожної з яких становить O_C , а сумарний обсяг усіх n часток дорівнює $O_{\Sigma}^6 = nO_C$. Тому відносно зменшення обсягу запропонованих схем, порівняно з відомими пороговими підходами, обчислюється за формулою:

$$\frac{O_{\Sigma}^{3,c}}{O_{\Sigma}^6} = \frac{(n-k)O_C}{nO_C} = \frac{n-k}{n}.$$

Отже, можемо зробити висновок, що запропоновані методи розподілу секрету за (k,n) схемою забезпечують у $\frac{n-k}{n}$ разів менший сумарний обсяг індивідуальних часток порівняно з відомими підходами. Тому потрібен менший обсяг пам'яті для зберігання розподіленого зображення і забезпечується менший час передавання цих часток.

2.5.1 Оцінка обсягу з попереднім ущільненням зображення

Для захищеної обробки та передавання зображень зменшення сумарного обсягу часток має практичне значення за умов обмеженої ємності пам'яті вузлів, низької пропускної здатності каналів зв'язку або необхідності скорочення часу

передавання даних. У таких випадках отримане вище зменшення сумарного обсягу індивідуальних часток може бути доповнене попереднім ущільненням зображення перед шифруванням і формуванням часток.

Попереднє ущільнення впливає на оцінку обсягу через зміну розміру подання зображення, яке подається на вхід криптографічного перетворення. Якщо після ущільнення обсяг вхідного подання зменшується з N до N_c , то коефіцієнт ущільнення становить $K_c = \frac{N}{N_c}$. За відсутності попереднього ущільнення $K_c = 1$, у разі зменшення обсягу вхідного подання $K_c > 1$.

У роботі [90] наведено аналіз алгоритмів ущільнення зображень із втратами, зокрема підходів на основі колірної субдискретизації, квантування кольорів, дискретного косинусного перетворення, вейвлет-перетворення та фрактального ущільнення. У роботі [91] досліджено ущільнення растрового подання зображення із використанням двовимірних ортогональних перетворень. Дискретне косинусне перетворення в цьому випадку відповідає JPEG-подібній схемі кодування, а перетворення Уолша-Адамара розглядається як обчислювально простіший варіант, який може забезпечувати близький рівень ущільнення для окремих типів зображень.

Для застосування в межах запропонованих схем метод, розглянутий у [91], застосовується до піксельного подання зображення: зображення поділяється на блоки, до них застосовується двовимірне ортогональне перетворення, після чого отримані коефіцієнти групуються та кодуються. Отже, результат такого етапу може бути поданий як байтовий вектор меншого обсягу, який надалі шифрується та розподіляється за запропонованою схемою.

У цій роботі показано, що групування коефіцієнтів після двовимірного ортогонального перетворення може зменшувати обсяг файлу з ущільненими коефіцієнтами приблизно у 1,2-2,5 рази порівняно з варіантом без групування. Це

значення характеризує додатковий ефект від організації коефіцієнтів у межах розглянутого способу ущільнення. Фактичне значення K_c визначається типом зображення, його початковим поданням, вибраним перетворенням, параметрами квантування, правилом групування коефіцієнтів та способом подальшого кодування.

Оскільки у попередніх оцінках сумарний обсяг індивідуальних часток визначається лінійно відносно обсягу вхідного подання, заміна N на $N_c = N / K_c$ забезпечує додаткове зменшення обсягу часток у K_c разів. Якщо без попереднього ущільнення сумарний обсяг часток запропонованого підходу становить $(n - k)O_c$, то з урахуванням попереднього ущільнення цей показник відносно початкового обсягу набуває вигляду $\frac{n - k}{K_c} O_c$.

Запропонована схема розподілу зменшує обсяг даних, які потрібно зберігати на окремих вузлах і передавати між учасниками відновлення. Попереднє ущільнення зображення підсилює цей ефект, оскільки ще до шифрування зменшує розмір вхідного подання на значення коефіцієнту K_c . У поєднанні ці два механізми знижують вимоги до обсягу пам'яті вузлів, пропускної здатності каналів і часу передавання часток, що є важливим для розподілених систем із ресурсними обмеженнями.

2.6 Оцінки складності реалізації методів шифрування та розподілу зображення

2.6.1 Оцінка складності реалізації зашифрування та розшифрування

Складність реалізації запропонованого методу шифрування оцінюється кількістю операцій, що здійснюються над байтовою послідовністю зображення під час виконання передбачених процедур перетворення. Усі основні процедури

запропонованого методу передбачають послідовну обробку байтів вектора зображення або блоків, сформованих з цього вектору.

В першу чергу варто додатково розділити сам процес шифрування (або будь-який інший етап) на два блоки: підготовчий етап (ініціалізація) та безпосередньо саме виконання заданих перетворень.

Тому спочатку необхідно ініціалізувати початковий стан РЗЛЗЗ, кількість та розміри піддіапазонів, ініціалізувати лічильники в межах кожного піддіапазону. Якщо приймати кожен ініціалізацію як одну операцію, то для присвоєння початкового стану РЗЛЗЗ – 1 операція, визначення розміру піддіазону як 1 операція, аналогічно ініціалізація по одному лічильнику для кожного піддіазону l операцій. Отже, кількість операцій потрібних для ініціалізації процесу шифрування визначається як:

$$T_{iniц} = l + 2.$$

Оскільки в межах даної роботи процес шифрування розглядається у двох варіантах реалізації – SP та PS, то доцільно оцінити складність реалізації кожного з цих варіантів окремо. При цьому основні процедури перетворення побудовані однаковим чином для обох варіантів, тому спочатку оцінимо складність таких перетворень як виконання підстановки та перестановки.

Для виконання підстановки необхідно обчислити значення маски Z_t . Для цього на кожному кроці t зі стану РЗЛЗЗ (R_t) отримуються два байтові значення X_t та Y_t . X_t визначається шляхом накладання маски 0xFF на поточний стан R_t , для того щоб виділити 8 молодших розрядів послідовності. Для того щоб визначити Y_t поточний стан регістру зсувається на 8 позицій вправо і аналогічно накладається маска 0xFF для отримання наступних 8 бітів. На основі отриманих значень X_t та Y_t значення Z_t обчислюється як $Z_t = aX_t + bY_t + c$, для чого потрібно виконати 4 операції. Таким чином обчислення маски Z_t вимагає виконання $T_Z = 7$ операцій.

Додатково накладання маски на байт включає зчитування поточного байта, додавання Z_t і запис результату, що потребує ще 3 операцій.

Таким чином загальна кількість операцій для виконання підстановки дорівнює:

$$T_{nidcm} = T_Z + 3 = 10.$$

Також спільною для обох режимів є процедура формування позиції перестановки. Вона включає оновлення стану РЗЛЗЗ шляхом виконання 3 операцій. Зчитування послідовності бітів для визначення номеру піддіапазону – 1 операція. Перевірка доступності піддіапазону, зчитування байту, який потрібно переставити, запис у новий вектор, оновлення стану лічильника додають ще 4 операції. Тому для формування перестановки потрібно виконати $T_{перест} = 8$ операцій.

Для шифрування за варіантом SP підстановка та перестановка виконуються як два окремі послідовні етапи. Оскільки, для обробки всього вектора зображення необхідно виконати N кроків, тому загальна кількість операцій для підстановки і перестановки складає:

$$T_{шифр}^{SP}(N) = T_{ініц} + N(T_{nidc} + T_{перест}) + 2N = 20N + l + 2 \approx 20N.$$

Тут також враховано, що для SP варіанту використовується додатковий, проміжний вектор і виконання зчитування з даного вектору та запис у зашифрований додає ще $2N$ операцій.

У варіанті PS підстановка і перестановка виконуються в одному циклі, тому складність реалізації зашифрування становить:

$$T_{шифр}^{PS}(N) = T_{ініц} + N(T_{nidc} + T_{перест}) = 18N + l + 2 \approx 18N.$$

Реалізація процесу розшифрування вимагає такої ж кількості операцій як і для зашифрування.

Для порівняння складності реалізації розглянуто методи, що відображають основні підходи до криптографічного перетворення зображень: класичне потокове

шифрування на основі блокового алгоритму, хаотичні методи перестановки та дифузії, методи з операціями у скінченному полі, методи з DNA-кодуванням, обробкою бітових площин та перетвореннями області подання зображення. Такий набір дозволяє зіставити запропонований метод із підходами, у яких захист зображення забезпечується різними за складністю механізмами зміни значень і позицій елементів.

Метод AES-128-CTR [92] розглядається як базовий криптографічний варіант, у якому для формування ключового потоку використовуються раунди AES-128, а зашифрування виконується шляхом накладання сформованого потоку на байти зображення. Метод Logistic-permutation-XOR [93] поєднує перестановку блоків або елементів зображення з накладанням маски, сформованої на основі логістичного відображення. Метод Arnold-Henon [94] використовує відображення Арнольда для перестановки позицій елементів та відображення Хенона для формування керувальних послідовностей дифузії. Метод $GF(2^8)$ -chaos [95] передбачає хаотичне керування перетвореннями та виконання операцій над байтовими значеннями у скінченному полі $GF(2^8)$, що суттєво збільшує кількість елементарних операцій.

До порівняння також включено метод Dynamic-DNA-chaos [96], у якому зображення перетворюється за допомогою хаотично керованих правил DNA-кодування та дифузії, метод СВР-WBPCD [97], що поєднує хаотичну блокову перестановку з ланцюговою дифузією зважених бітових площин, а також метод IWT-hyperchaos-Zhong2022 [98], у якому застосовується цілочисельне вейвлет-перетворення та гіперхаотичне формування керувальних послідовностей. Ці методи відрізняються кількістю проходів по даних, наявністю додаткових перетворень подання зображення, складністю генерації керувальних послідовностей та типом операцій, що виконуються над елементами зображення.

Для узгодженого порівняння всі методи приведено до кількості базових операцій на один байт вхідного подання (в обчислення не було включено

внутрішню реалізацію базових математичних операцій, тощо). Коефіцієнт складності визначається відносно запропонованого PS-варіанта, для якого оцінка становить 18 операцій на один байт. Результати наведено в табл. 2.4.

Таблиця 2.4 – Порівняльна оцінка складності реалізації з відомими методами шифрування

Метод	Кількість операцій на 1 байт	Коефіцієнт складності
AES-128-CTR	107	5,9
Logistic-permutation-XOR	157	8,7
Arnold-Henon	101	5,6
GF(2^8)-chaos	1074	59,6
Dynamic-DNA-chaos	151	8,4
CBP-WBPCD	249	13,8
IWT-hyperchaos-Zhong2022	190	10,5
Запропонований метод (SP)	20	1,1
Запропонований метод (PS)	18	1

З наведених оцінок видно, що запропонований PS-варіант має найменшу кількість операцій на один байт серед розглянутих методів. Найближчими за складністю є Arnold-Henon та AES-128-CTR, однак їхня реалізація вимагає у 5,6 та 5,9 рази більше операцій ніж PS-варіант. Методи Logistic-permutation-XOR і Dynamic-DNA-chaos потребують у 8,7 та 8,4 рази більше операцій, що пов'язано з додатковим формуванням хаотичних послідовностей, перестановками, XOR-дифузєю або DNA-перетвореннями. Найбільшу складність має GF(2^8)-chaos, для якого оцінка становить 1074 операції на байт, тобто у 59,6 рази більше за запропонований PS-варіант.

2.6.2 Оцінка складності реалізації методів розподілу та відновлення

Для узагальненого коаліційного методу структура розміщення блоків у частках вважається визначеною на рівні побудови системи для заданих параметрів k та n . Тому під час розподілу конкретного зашифрованого вектору C не

виконується повторне формування коаліційних множин або індексних відповідностей. Оцінюються лише операції, які безпосередньо виконуються над байтами вектора C , частками учасників і часткою дилера.

На основному етапі кожен байт зчитується з вектору C один раз і записується до k часток учасників відповідно до заданої структури розподілу. Отже, для одного байту виконується одна операція зчитування та k операцій запису. Для всього вектору обсягом N байтів складність розміщення даних у частках учасників становить:

$$T_{розп}^{yz} = N(k + 1).$$

Частка дилера формується в тому самому проході, що й розміщення байтів у частках учасників. Для цього поточний байт, який уже зчитано з C , одночасно використовується для послідовного обчислення частки дилера за рахунок виконання операції додавання за модулем 2. Оскільки повторне зчитування цього байту не виконується, для кожного байту додатково враховується лише операція обчислення суми та запис оновленого значення. Тому складність формування частки дилера становить $T_{дил}^{yz} = 2N$ операцій.

Отже, загальна складність реалізації розподілу за узагальненим коаліційним методом дорівнює:

$$T^{yz} = T_{розп}^{yz} + T_{дил}^{yz} = N(k + 1) + 2N = N(k + 3).$$

Під час відновлення для узагальненого методу отримувач може мати k або більше часток (до $k \leq n$). Також він отримує значення частки дилера D та індекси часток, і блоків у межах цих часток. Основне завдання отримувача при відновленні полягає в тому, щоб послідовно зчитати отримані частки, визначити унікальні наявні блоки і за необхідності відновити блок якого не вистачає за рахунок частки дилера.

Оскільки один і той самий блок може входити до кількох отриманих часток,

під час відновлення використовується допоміжний масив ознак обробки блоків. Кожна позиція такого масиву відповідає кожному послідовному блоку m . Відповідно унікальність буде звірятись згідно отриманих індексів блоків і оновлюватись у визначеній позиції масиву (який початково ініціалізується нулями), якщо блок вже отриманий (записаний) значення змінюється на 1, якщо ні то залишається 0. Така перевірка дозволяє забезпечити з одного боку зменшення кількості проходів по всім даним і також корекність обчислення суми за модулем 2 усіх блоків, коли існує необхідність задіяти частку дилера у відновленні.

Якщо для відновлення доступні всі n часток, частка дилера не використовується, оскільки всі блоки вже наявні в отриманих частках згідно зі схемою розподілу. У цьому випадку отримувач послідовно обробляє службові індекси блоків у частках і записує їх у відновлений вектор без повторень. Так як визначити на якому кроці зчитування блоків всі m блоків будуть однозначно доступні не є можливим, для оцінки кількості операцій використаємо повний перебір усіх блокових входжень у отриманих частках. Кількість перевірок службових індексів становить $n(n-k)$ операцій. Побайтова обробка виконується лише для унікальних блоків, які вперше зустрічаються під час зчитування часток. Їх сумарний обсяг дорівнює N . Для кожного такого байту враховується зчитування з частки та запис у відповідну позицію відновлюваного вектора. Тому кількість операцій виділених на зчитування та запис усіх байтів становить $2N$.

Отже, операційна складність відновлення за наявності всіх n часток становить:

$$T_{відн}^{yz} = n(n-k) + 2N$$

Якщо для відновлення доступні рівно k часток, то після їх обробки залишиться один відсутній блок, який необхідно буде відновити з використанням частки дилера. При такому варіанті для перевірки наявних блоків потрібно виконати $k(n-k)$ операцій. З урахуванням відсутності одного блоку загальний

обсяг даних, які обробляються при відновленні становить $N - \frac{N}{n}$, де $\frac{N}{n}$ – кількість байтів відсутнього блоку.

Для наявних блоків враховується зчитування, запис у відновлюваний вектор і додавання за модулем 2 до проміжного значення. Для відсутнього блоку враховується додавання за модулем 2 частки дилера D з цим проміжним значенням і запис результату. В сумі ця складова становить $4N$ операцій. Тому операційна складність відновлення за наявності рівно k часток:

$$T_{відн}^{y3} = k(n - k) + 4N.$$

Розглянемо оцінку складності реалізації для спрощеного методу розподілу за схемою (k, n) з циклічним перекриттям блоків. У даній реалізації також приймається, що система побудована вже з заданим правилом вибору індексів перестановки згідно з параметрами k та n .

На відміну від узагальненого коаліційного методу, у спрощеній схемі вектор C розглядається як послідовність із n блоків, а кожен блок відповідно до заданого правила записується до $n-k$ часток учасників. Тому для кожного байту виконується одне зчитування з вектору C та $n-k$ операцій запису до часток. Відповідно, кількість операцій для розміщення байтів у частках учасників становить:

$$T_{част}^{cn} = n(n - k + 1).$$

Аналогічно як і для узагальненого методу розподілу формується й частка дилера як сума за модулем 2 усіх блоків (байтів) зашифрованого вектору та додає $2N$ операцій до загальної складності. Таким чином кількість операцій для виконання процесу розподілу дорівнює:

$$T_{розп}^{cn} = N(n - k + 1) + 2N = N(n - k + 3)$$

Оцінка складності відновлення для узагальненого коаліційного та спрощеного методів збігається, оскільки в обох випадках отримувач виконує однакову послідовність дій: обробляє індекси блоків у отриманих частках, визначає

унікальні блоки за допомогою масиву ознак, записує їх у відповідні позиції відновлюваного вектора, а за мінімальної кількості часток додатково використовує частку дилера для обчислення одного відсутнього блока.

Таким чином, у підрозділі виконано оцінку складності реалізації основних етапів запропонованих методів: зашифрування, розшифрування, розподілу та відновлення зображення. Для зашифрування окремо визначено складність підготовчого етапу, формування маскувального значення, підстановки та перестановки для режимів SP і PS. Отримані оцінки показують, що режим PS має меншу складність реалізації порівняно з SP, оскільки не передбачає формування проміжного результату після підстановки та його подальшого повторного використання під час перестановки. Розшифрування має складність, співмірну з PS-режимом, оскільки позиції перестановки відтворюються за тим самим правилом, а пряма підстановка замінюється оберненою.

Для кількісного порівняння операцій розподілу узагальнений метод прийнято як базовий рівень 100%. Тому для схеми (3,5) спрощений метод потребує 83.3% операцій відносно узагальненого, тобто на 16.7% менше. Для схеми (3,6) обидва методи мають однакову оцінку, а для (2,5) спрощений метод потребує 120% операцій відносно узагальненого, тобто на 20% більше.

2.8 Висновки до другого розділу

У другому розділі розроблено методи шифрування та розподілу секретного вмісту зображення, у яких початкові дані подаються у вигляді байтової послідовності, що надалі перетворюється у зашифрований вектор. Такий підхід дає змогу розглядати обробку зображення як послідовність взаємопов'язаних етапів: початкове подання даних, зашифрування, формування часток, відновлення зашифрованого вектора та зворотне отримання вхідного зображення.

Запропоновано метод шифрування, що поєднує операції підстановки та

перестановки. Підстановка забезпечує зміну значень байтів, а перестановка — зміну їхніх позицій у межах усього зашифрованого вектора. Для формування позицій перестановки використано РЗЛЗЗ, поділ простору індексів на піддіапазони та систему лічильників. Така організація забезпечує виконання перестановки псевдовипадковим чином з однозначно визначеним алгоритмом за рахунок використання секретного ключа.

Для методу шифрування розглянуто два варіанти реалізації — SP та PS. У варіанті SP підстановка і перестановка виконуються послідовно з формуванням проміжного результату, тоді як у варіанті PS ці операції поєднуються в одному проході. Теоретична оцінка складності показала, що PS-варіант потребує 18 базових операцій на один байт, а SP-варіант — 20 базових операцій на один байт. Порівняння з відомими методами шифрування зображень показало, що для розглянутих підходів складність становить від 101 до 1074 операцій на байт, тому PS-варіант потребує у 5,6–59,6 рази менше базових операцій для реалізації.

Удосконалено метод розподілу секретного вмісту зображення за (n,n) -схемою за рахунок попереднього шифрування. Також запропоновано два варіанти порогового розподілу за (k,n) -схемою: узагальнений метод з використанням коаліцій учасників і спрощений метод із перекриттям блоків. В обох випадках об'єктом розподілу є не відкрите зображення, а зашифрований вектор. Для порогових схем передбачено використання частки дилера, яка передається окремо та застосовується під час відновлення для обчислення відсутнього блоку.

Встановлено, що запропоновані (k,n) -схеми зменшують сумарний обсяг індивідуальних часток, які необхідно зберігати на вузлах або передавати між учасниками відновлення. Для схеми $(3,5)$ сумарний обсяг індивідуальних часток становить 40% від відповідного обсягу аніж при передачі часток у повному обсязі. У разі попереднього ущільнення зображення цей показник знижується до 33,3...16% в залежності від значення коефіцієнта ущільнення K_c , що підтверджує

доцільність поєднання попереднього ущільнення з запропонованими схемами розподілу для систем з обмеженою пам'яттю вузлів і пропускнуою здатністю каналів.

Отримані оцінки алгоритмічної складності показали, що запропонований PS-варіант шифрування є найменш витратним серед розглянутих методів: він потребує 18 базових операцій на байт, тоді як SP-варіант потребує 20 операцій на байт, а відомі методи шифрування зображень – від 101 до 1074 операцій на байт. Отже, PS-варіант потребує у 5,6...59,6 рази менше операцій порівняно з розглянутими відомими методами.

Для процедур розподілу встановлено, що ефективність коаліційної та спрощеної (k,n) -схем залежить від вибраних параметрів k і n . Для схеми (3,5) спрощений метод потребує 83,3% операцій відносно коаліційного, для (3,6) обидва методи мають однакову оцінку. Для процедур відновлення обидві (k,n) -схеми мають однакову оцінку складності, тому відмінність між ними проявляється саме на етапі формування часток.

3 ЗАСОБИ РЕАЛІЗАЦІЇ МЕТОДІВ ШИФРУВАННЯ ТА РОЗПОДІЛУ ЗОБРАЖЕНЬ

3.1 Структура даних і програмних компонентів

Програмна реалізація запропонованих процедур будується як послідовність обчислювальних компонентів, що виконують перетворення байтового вмісту зображення, формування часток, відновлення зашифрованих даних і подальше розшифрування. На рівні обчислювального ядра основними об'єктами є байтовий вектор, параметри ключа, фіксована конфігурація РЗЛЗЗ, стан лічильників, зашифрований вектор, блоки зашифрованого вектора, частки учасників і частка дилера.

Базовою структурою для подання даних є байтовий буфер поданий на рис. 3.1:

```
struct ByteBuffer
{
    vector<uint8_t> data;
};
```

Рисунок 3.1 – Фрагмент коду ініціалізації байтового буферу для збереження даних зображення

Структура `ByteBuffer` використовується для подання вхідного вектора V , зашифрованого вектора C , відновленого зашифрованого вектора та результату розшифрування. Параметри, що використовуються під час шифрування та розшифрування, подаються структурою (рис. 3.2):

```
struct KeyParams
{
    uint64_t K1;
    int r;
    vector<int> direction_bits;
    SubstitutionParams substitution;
};
```

Рисунок 3.2 – Фрагмент коду ініціалізації ключових параметрів

Поле k_1 задає початковий стан 64-бітового РЗЛЗЗ. Значення r визначає

кількість бітів псевдовипадкової послідовності, з яких формується номер лічильника під час вибору позиції перестановки. Масив `direction_bits` задає напрям руху кожного лічильника. Структура `SubstitutionParams` містить параметри прямої та оберненої підстановки байтів.

Конфігурація РЗЛЗЗ задається на рівні реалізації генератора. Для цього фіксується розрядність регістра та правило зворотного зв'язку, подане у вигляді бітової маски або еквівалентного полінома (рис. 3.3):

```
struct LFSRConfig
{
    static constexpr int register_size = 64;
    static constexpr uint64_t feedback_mask = FEEDBACK_MASK;
};
```

Рисунок 3.3 – Фрагмент коду ініціалізації параметрів РЗЛЗЗ

Бітова маска `feedback_mask` визначає розряди регістра, що беруть участь в обчисленні нового біта через операцію XOR. Початковий стан генератора задається значенням `K1`, після чого генератор формує псевдовипадкову послідовність.

Компонент генератора ПВП має такий узагальнений інтерфейс (рис. 3.4):

```
class LFSRGenerator
{
public:
    void Init(uint64_t K1);

    uint8_t NextBit();
    uint8_t NextMask();
    int NextCounterIndex(int r);
};
```

Рисунок 3.4 – Інтерфейс генератора ПВП на основі РЗЛЗЗ

Метод `NextBit()` виконує один крок РЗЛЗЗ і повертає черговий біт псевдовипадкової послідовності. Метод `NextMask()` формує маскувальний байт для підстановки. Метод `NextCounterIndex(r)` формує число з r послідовних бітів. Це число використовується як номер лічильника, з якого буде отримано поточну абсолютну позицію запису або зчитування.

Стан перестановки в реалізації задається масивом лічильників (рис. 3.5):

```

struct Counter
{
    int position;
    int step;
    int remaining;
};

struct CounterState
{
    vector<Counter> counters;
};

```

Рисунок 3.5 – Опис структури лічильників

Поле `position` містить поточну абсолютну позицію у векторі. Поле `step` задає напрям руху лічильника: `+1` для прямого напрямку та `-1` для зворотного. Поле `remaining` визначає кількість позицій, доступних для цього лічильника.

Ініціалізація лічильників виконується перед початком зашифрування або розшифрування (рис. 3.6):

```

CounterState InitCounters(int N, int r, vector<int> direction_bits)
{
    int L = 1 << r;

    CounterState state;
    state.counters.resize(L);

    for (int beta = 0; beta < L; beta++) {
        int start = (beta * N) / L;
        int end = ((beta + 1) * N) / L;
        int size = end - start;

        if (direction_bits[beta] == 0) {
            state.counters[beta].position = start;
            state.counters[beta].step = +1;
        }
        else {
            state.counters[beta].position = end - 1;
            state.counters[beta].step = -1;
        }

        state.counters[beta].remaining = size;
    }

    return state;
}

```

Рисунок 3.6 – Фрагмент коду ініціалізації початкових станів лічильників

У цій структурі піддіпазони використовуються тільки для початкового визначення параметрів відповідного лічильника. Після ініціалізації подальша робота процедури перестановки виконується через масив лічильників. Генератор ПВП формує номер лічильника, а механізм отримання позиції та оновлення його стану розглядається в підрозділі, присвяченому реалізації процедур зашифрування

та розшифрування.

Компонент зашифрування приймає вхідний байтовий буфер і параметри ключа, після чого формує зашифрований буфер (рис. 3.7):

```
class EncryptionCore
{
public:
    ByteBuffer Encrypt(const ByteBuffer& input,
                      const KeyParams& key);
};
```

Рисунок 3.7 – Фрагмент коду ініціалізації процедури зашифрування

У межах цього компонента виконуються підстановка значень байтів і запис результатів у нові позиції. Варіанти виконання SP і PS подаються в наступному підрозділі як дві реалізації цього компонента.

Компонент розшифрування має інтерфейс, аналогічний зашифруванню (рис. 3.8).

```
class DecryptionCore
{
public:
    ByteBuffer Decrypt(const ByteBuffer& encrypted,
                      const KeyParams& key);
};
```

Рисунок 3.8 – Фрагмент коду ініціалізації процедури розшифрування

Його задача полягає у повторному формуванні тієї самої послідовності маскувальних значень і позицій, що використовувалася під час зашифрування, з подальшим застосуванням оберненої підстановки.

Для формування часток зашифрований вектор подається у вигляді послідовності блоків. Службова інформація про блокове подання зберігається окремою структурою (рис. 3.9):

```
struct BlockLayout
{
    int block_count;
    int block_size;
    int original_size;
};
```

Рисунок 3.9 – Опис структури блоків розподілених даних зображення

Поле `block_count` задає кількість блоків, поле `block_size` — розмір одного блока після вирівнювання, поле `original_size` — довжину зашифрованого вектору **C** до службового доповнення. Значення `original_size` використовується після відновлення для відкидання доданих байтів останнього блока.

Окремий блок подається структурою (рис. 3.10):

```
struct Block
{
    int index;
    vector<uint8_t> data;
};
```

Рисунок 3.10 – Опис структури одного блоку розподілених даних зображення

Поле `index` задає початковий номер блока у зашифрованому векторі. Поле `data` містить байтовий вміст блока. Наявність індексу дозволяє відновлювати порядок блоків незалежно від порядку надходження часток.

Частка учасника подається як набір блоків (рис. 3.11):

```
struct Share
{
    int participant_index;
    vector<Block> blocks;
};
```

Рисунок 3.11 – Опис структури розподіленої частки зображення

Поле `participant_index` задає номер учасника, а `blocks` містить блоки, що входять до складу відповідної частки. Параметри k та n передаються до процедур розподілу і відновлення як параметри виклику.

Частка дилера подається окремою структурою (рис. 3.12):

```
struct DealerShare
{
    vector<uint8_t> data;
};
```

Рисунок 3.12 – Опис структури частки дилера

Поле `data` містить байтовий вектор **D**, сформований операцією XOR над блоками зашифрованого вектора. Цей вектор використовується у процедурі відновлення одного відсутнього блоку.

Результат роботи компонента розподілу подається структурою (рис. 3.13):

```
struct SplitResult
{
    BlockLayout layout;
    vector<Share> shares;
    DealerShare dealer;
};
```

Рисунок 3.13 – Опис структури компонентів структури розподілу зображення

Структура `SplitResult` об'єднує службовий опис блокового подання, частки учасників і частку дилера. Параметри k , n визначаються на рівні виклику відповідної процедури розподілу.

За формування часток відповідає клас `SharingCore`, його структура наведена на рис. 3.14.

```
class SharingCore
{
public:
    SplitResult Split(const ByteBuffer& encrypted,
                     int k,
                     int n);
};
```

Рисунок 3.14 – Структура класу `SharingCore` для формування часток

Для коаліційного методу внутрішня реалізація формує індексні множини допустимих коаліцій і розміщує блоки у частках відповідно до цих множин. Для спрощеного методу внутрішня реалізація використовує циклічне призначення блоків. Деталізація цих двох процедур подається в окремих підрозділах.

Компонент відновлення приймає доступні частки учасників, частку дилера, службову інформацію про блокове подання і параметри схеми що показано на рис. 3.15.

```
class RecoveryCore
{
public:
    ByteBuffer Restore(const vector<Share>& available_shares,
                      const DealerShare& dealer,
                      const BlockLayout& layout,
                      int k,
                      int n);
};
```

Рисунок 3.15 – Компонент процедури відновлення зашифрованого вектору

Під час відновлення, блоки з доступних часток розміщуються за їхніми

індексами. Якщо всі блоки наявні, вони об'єднуються у правильному порядку. Якщо відсутній один блок, він обчислюється з використанням частки дилера. Після об'єднання блоків формується результат за розміром `original_size`.

Відповідність між програмними компонентами та основними даними наведено на рис. 3.16. Ця структура відображає фактичний порядок взаємодії між компонентами запропонованої реалізації методів шифрування та розподілу секретного вмісту зображення.

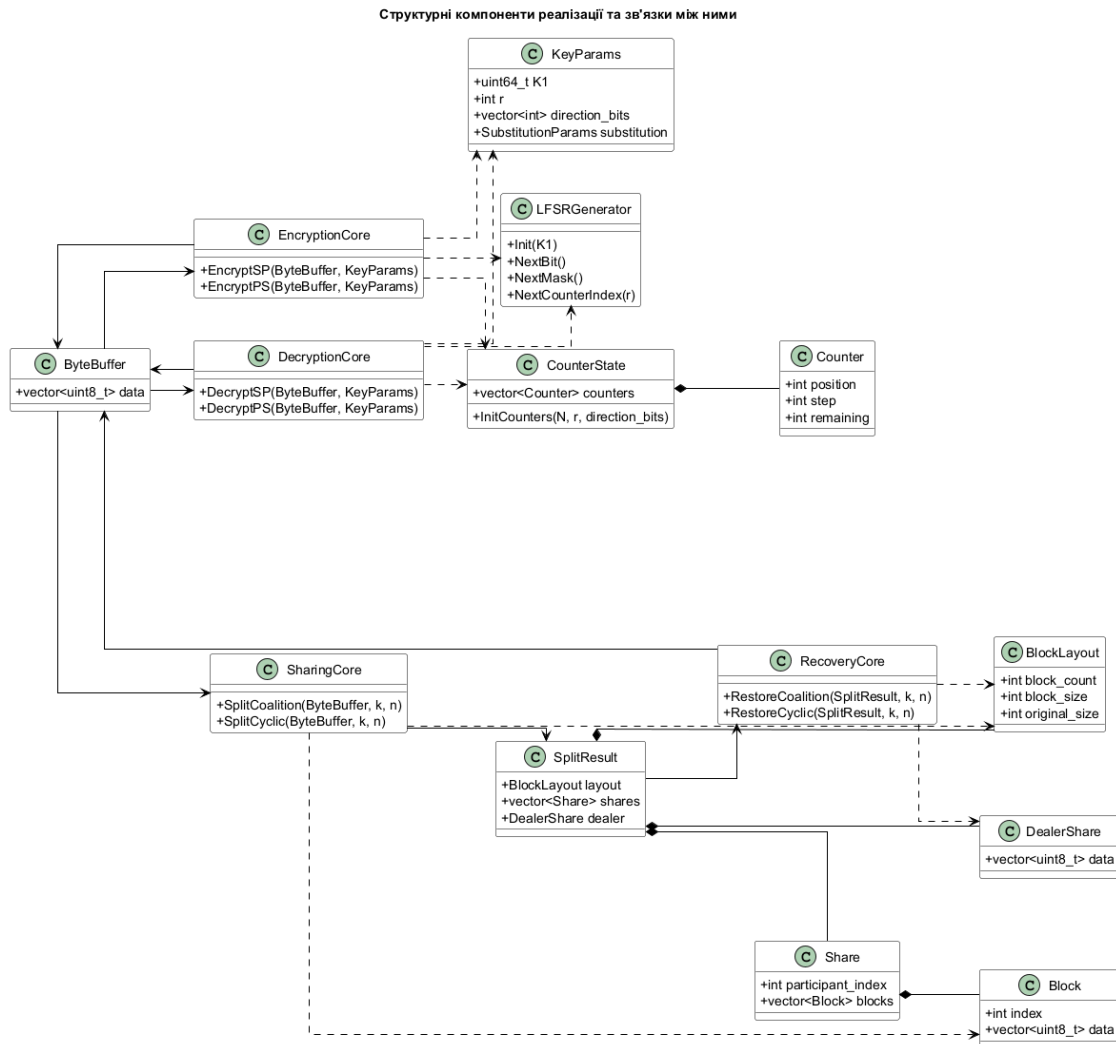


Рисунок 3.16 – UML-діаграма структур даних і основних компонентів реалізації

3.2 Програмний засіб для зашифрування та розшифрування зображення

Процедури зашифрування та розшифрування реалізуються як обробка байтового вектора з використанням двох основних операцій: підстановки значення байту та перестановки позиції запису або зчитування. Генератор ПВП на основі РЗЛЗЗ формує два типи значень: маскувальний байт для підстановки та номер лічильника, з якого отримується абсолютна позиція у векторі. Стан перестановки задається масивом лічильників, ініціалізованих відповідно до довжини вектора N , параметра r та напрямів руху, визначених ключем.

Один крок РЗЛЗЗ повертає біт псевдовипадкової послідовності. Для підстановки формується байт γ , отриманий з восьми послідовних бітів генератора, ця процедура наведена на рис. 3.17.

```
uint8_t NextMask(LFSRGenerator& R)
{
    uint8_t gamma = 0;

    for (int i = 0; i < 8; i++) {
        gamma = (gamma << 1) | R.NextBit();
    }

    return gamma;
}
```

Рисунок 3.17 – Процедура формування наступного значення генератора

Для вибору позиції перестановки генератор формує номер лічильника B з r послідовних бітів за процедурою, що наведена на рис. 3.18.

```
int NextCounterIndex(LFSRGenerator& R, int r)
{
    int beta = 0;

    for (int i = 0; i < r; i++) {
        beta = (beta << 1) | R.NextBit();
    }

    return beta;
}
```

Рисунок 3.18 – Процедура генерування номера лічильника

Значення β не є позицією байту у векторі. Воно визначає тільки номер лічильника, а номер позиції визначається з урахуванням поточного стану

відповідного лічильника, за процедурою зображеною на рис. 3.19.

```

int NextPosition(LFSRGenerator& R,
                CounterState& state,
                int r)
{
    while (true) {
        int beta = NextCounterIndex(R, r);

        if (state.counters[beta].remaining > 0) {
            int p = state.counters[beta].position;

            state.counters[beta].position += state.counters[beta].step;
            state.counters[beta].remaining--;

            return p;
        }
    }
}

```

Рисунок 3.19 – Процедура визначення номеру позиції для перестановки

Отже, перестановка реалізується без збереження повної таблиці позицій. Послідовність позицій формується під час проходження по вектору за рахунок стану РЗЛЗЗ і масиву лічильників. Якщо вибраний лічильник уже вичерпав доступні номери, виконується формування наступного номеру лічильника.

3.2.1 Програмна реалізація шифрування за SP варіантом

У варіанті SP спочатку виконується підстановка, після чого застосовується перестановка. На першому етапі для кожного байту вектора V формується маскувальне значення γ , а результат підстановки записується до проміжного вектора V' згідно з процедурою наведеною на рис. 3.20.

```

EncryptSP(V, K)
{
    for (i = 0; i < N; i++) {
        gamma = Mask(R);
        Vp[i] = S(V[i], gamma, K.sub);
    }
    ctr = InitCounters(N, K.r, K.dir);

    for (i = 0; i < N; i++) {
        p = Pos(R, ctr, K.r);
        C[i] = Vp[p];
    }
    return C;
}

```

Рисунок 3.20 – Процедура зашифрування за SP варіантом

Розшифрування для SP виконується у зворотній послідовності. Спочатку відтворюється така сама послідовність визначення позицій перестановки, і байти з

C повертаються у відповідні позиції проміжного вектора V' . Далі до V' застосовується обернена підстановка, наведена на рис. 3.21.

```

DecryptSP(C, K)
{
    for (i = 0; i < N; i++) {
        p = Pos(Rpos, ctr, K.r);
        Vp[p] = C[i];
    }
    for (i = 0; i < N; i++) {
        gamma = Mask(Rmask);
        V[i] = S_inv(Vp[i], gamma, K.sub);
    }

    return V;
}

```

Рисунок 3.21 – Процедура розшифрування за SP варіантом

На рис. 3.22 зображена схема алгоритму зашифрування за варіантом SP.

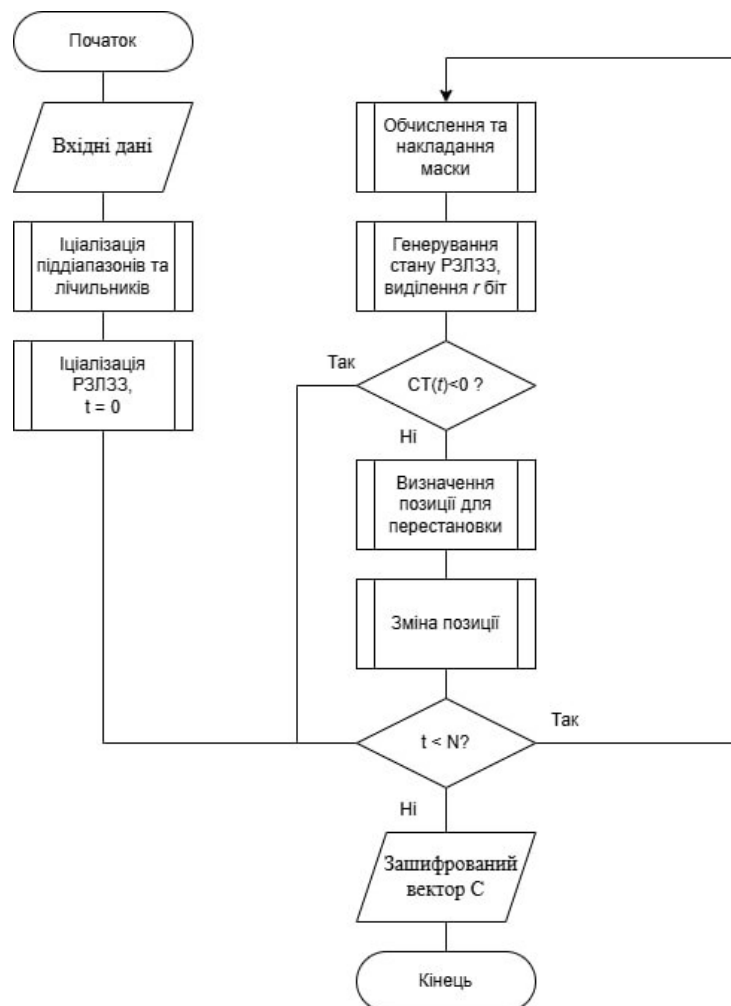


Рисунок 3.22 – Схема алгоритму зашифрування за варіантом SP

Усі процедури наведені на рис. 3.20 і рис. 3.21 є складовими цього алгоритму.

3.2.2 Програмна реалізація шифрування за PS варіантом

У варіанті PS спочатку визначається позиція байту за правилом перестановки, з вхідного вектора зчитується байт із цієї позиції, після чого до нього одразу застосовується підстановка і результат записується у вихідний вектор. Опис процедури зашифрування зображення наведено на рис. 3.23.

```
EncryptPS(V, K)
{
    R.Init(K.K1);
    N = V.size();

    C.resize(N);
    ctr = InitCounters(N, K.r, K.dir);

    for (i = 0; i < N; i++) {
        p = Pos(R, ctr, K.r);
        gamma = Mask(R);
        C[i] = S(V[p], gamma, K.sub);
    }

    return C;
}
```

Рисунок 3.23 – Процедура зашифрування за варіантом PS

Тут перестановка реалізується як вибір номера позиції p у вхідному векторі V . Після цього для байту $V[p]$ виконується підстановка і результат записується у поточну позицію вихідного вектора $C[i]$. Тобто у PS немає проміжного вектора V' , бо перестановка і підстановка поєднані в одному проході.

Розшифрування для PS повинне повторити той самий порядок формування позицій і маски. Оскільки під час зашифрування байт з позиції p вектору V після підстановки записувався у позицію i вектора C . Тому під час розшифрування результат оберненої підстановки з $C[i]$ переставляється у позицію p , ця процедура наведена на рис. 3.24.

```

DecryptPS(C, K)
{
    R.Init(K.K1);
    N = C.size();

    V.resize(N);
    ctr = InitCounters(N, K.r, K.dir);

    for (i = 0; i < N; i++) {
        p = Pos(R, ctr, K.r);
        gamma = Mask(R);
        V[p] = S_inv(C[i], gamma, K.sub);
    }
    return V;
}

```

Рисунок 3.24 – Процедура розшифрування за варіантом PS

Коректність розшифрування забезпечується однаковою ініціалізацією РЗЛЗЗ і лічильників, а також однаковим порядком викликів операцій підстановки та перестановки у процедурах зашифрування та розшифрування.

Рис. 3.25 подано у вигляді UML-діаграми послідовності, де по вертикалі відображено основних учасників процедури, а по горизонталі — повідомлення та виклики між ними. Послідовність дій читається згори вниз: спочатку передаються вхідні дані та параметри ключа, далі ініціалізуються службові структури, після чого в циклі виконується обробка елементів вхідного вектора. Окремі стрілки на діаграмі показують передавання керування між компонентами, відповідальними за формування керувальної послідовності, вибір піддіапазону, зміну лічильника та запис результату у вихідний вектор. Завершальним етапом є повернення сформованого зашифрованого вектора як результату виконання процедури.

3.3 Програмна реалізація коаліційного методу розподілу та відновлення за (k,n) -схемою

Коаліційний метод реалізується після формування зашифрованого вектора **C**. Вхідними даними програмного модуля є **C**, параметри k, n та сформовані індексні групи **G**. На етапі реалізації зашифрований вектор поділяється на блоки, після чого кожний блок записується до відповідних часток учасників разом із його індексом.

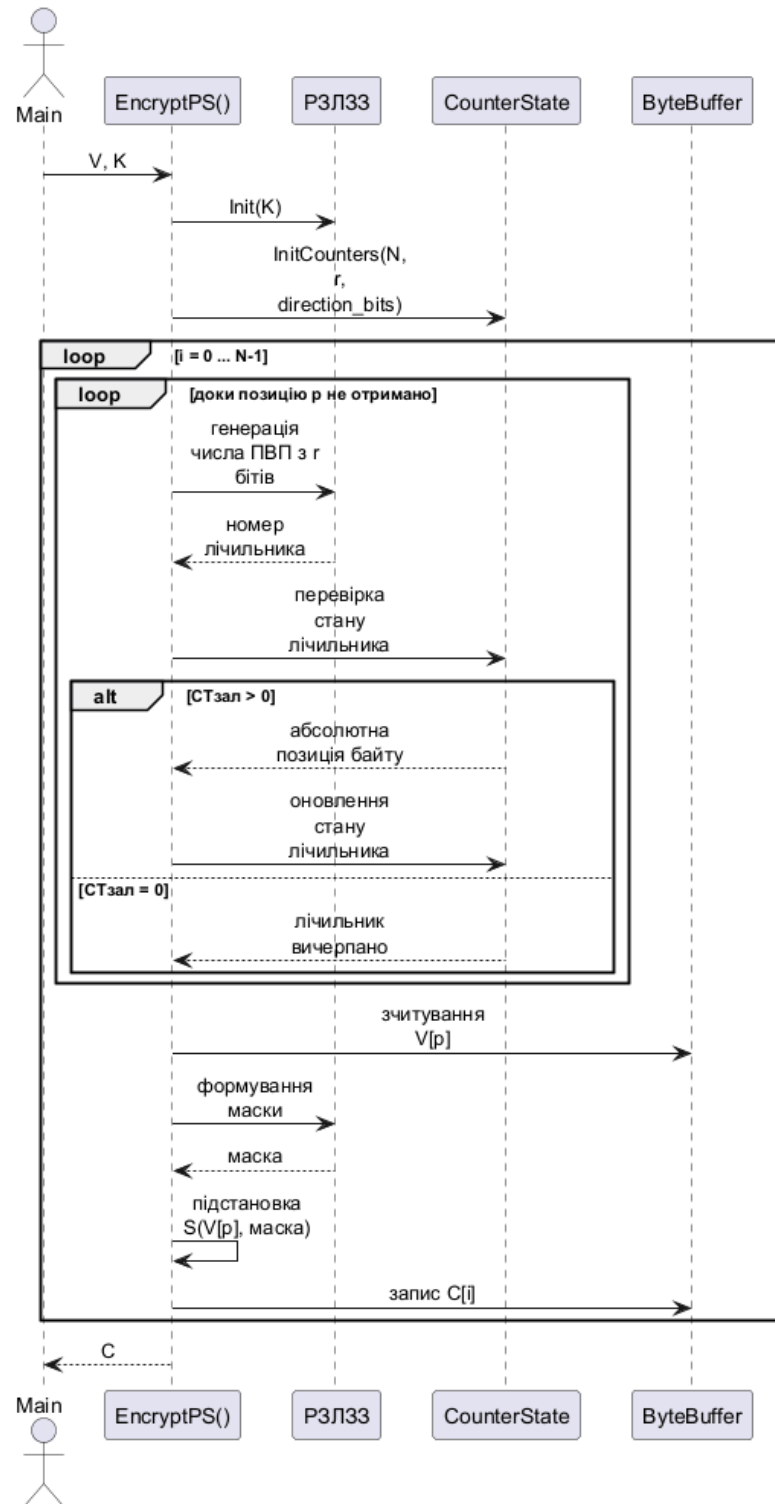


Рисунок 3.25 – UML-діаграма послідовності виконання процедури зашифрування за варіантом PS

Частка дилера D формується як побайтовий XOR усіх блоків. Процедура розподілу зображена на рис. 3.26.

```

SplitCoalition(C, k, n)
{
    G = BuildIndexGroups(k, n);
    B = SplitToBlocks(C, G.count);

    for each block B[j]:
        for each participant q:
            if q ∉ G[j]:
                P[q].Add(j, B[j]);

    D = B[1] XOR B[2] XOR ... XOR B[m];

    return P[1..n], D;
}

```

Рисунок 3.26 — Процедура розподілу за коаліційним методом

Метод `BuildIndexGroups(k, n)` формує групи індексів учасників, метод `SplitToBlocks` ділить зашифрований вектор на блоки, а операція `P[q].Add(j, B[j])` додає до частки учасника блок разом із його індексом. Індеси блоків потрібні для подальшого збирання зашифрованого вектора у правильному порядку. На рис. 3.27 наведена схема алгоритму розподілу зображення за коаліційним методом.

Відновлення виконується за доступною множиною часток. Під час зчитування блоків формується масив відновлених блоків і одночасно формується значення X як сума за модулем 2 усіх отриманих блоків без повторень. Якщо після цього відсутній один блок, його значення обчислюється без повторного проходу по всіх блоках за формулою $B_{\text{відсут}} = D \oplus X$.

Процедура відновлення зображена на рис. 3.28:

Параметр `received unique block` показує, що однаковий блок, який може повторюватися в кількох частках, враховується лише один раз. Після вставлення відсутнього блоку всі блоки об'єднуються за індексами, у результаті чого формується відновлений зашифрований вектор C . Якщо відсутніх блоків більше одного, наявний набір часток не забезпечує відновлення в межах цієї процедури.

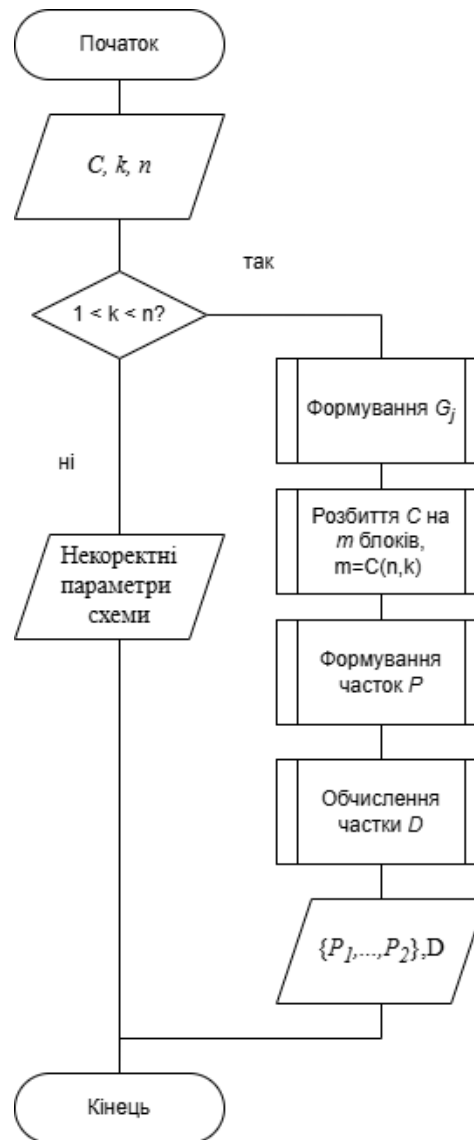


Рисунок 3.27 – Схема алгоритму розподілу секрету за коаліційним методом

```

RestoreCoalition(P_available, D)
{
  B[1..m] = empty;
  X = 0;

  for each received unique block (j, B[j]):
    insert B[j] by index;
    X = X XOR B[j];

  if one block is missing:
    B[miss] = D XOR X;

  C = JoinBlocks(B[1..m]);
  return C;
}

```

Рисунок 3.28 – Процедура відновлення зображення за коаліційним методом

3.4 Програмна реалізація спрощеного методу

Спрощений метод реалізується через попереднє формування набору відповідностей між індексами блоків і номерами часток. На відміну від коаліційного методу, тут не генеруються індексні групи учасників. Основним підготовчим етапом є визначення того, який блок, скільки разів і до яких саме часток має бути записаний. У програмній реалізації така відповідність подається як список пар: $(\text{block_index}, \text{share_index})$. Де block_index – індекс блока зашифрованого вектора, а share_index – номер частки, до якої цей блок записується. Такий список формується один раз для заданих параметрів k та n і використовується під час розподілу часток.

Процедура визначення відповідностей між блоками зашифрованого вектора та частками учасників зображена на рис. 3.29.

```
BuildBlockShareLinks(k, n)
{
    CheckParams(k, n);

    rho = n - k;
    links = empty;
    repeat[1..n] = 0;

    for q = 1 to n {
        for h = 0 to rho - 1 {
            j = ((q - 1 + h) mod n) + 1;

            links.Add(block_index = j,
                     share_index = q);

            repeat[j]++;
        }
    }
    CheckRepeatCount(repeat, rho);
    return links;
}
```

Рисунок 3.29 – Процедура визначення відповідності між блоками та частками розподілу

У цьому фрагменті rho визначає кількість блоків, з яких складається кожна частка. Для цього формується перелік індексів блоків, які мають бути до неї записані. Значення $\text{repeat}[j]$ використовується для контролю кількості включень кожного блоку до часток. Після визначення відповідностей зашифрований вектор ділиться на n блоків. Далі процедура проходить за списком links і виконується

додавання певного блоку до відповідної частки, ця процедура наведена на рис. 3.30.

```

SplitSimplified(C, k, n)
{
    blocks = SplitToBlocks(C, n);
    links = BuildBlockShareLinks(k, n);

    for each link in links {
        j = link.block_index;
        q = link.share_index;

        P[q].Add(j, blocks[j]);
    }

    D = XOR_All(blocks);

    return P[1..n], D;
}

```

Рисунок 3.30 – Процедура відновлення за спрощеним методом

Метод `SplitSimplified` виконує основний цикл розподілу та працює з попередньо сформованим набором відповідностей між блоками та частками. Кожна відповідність містить індекс блоку j та номер частки q . За цією парою з масиву `blocks` вибирається блок `blocks[j]` і записується до частки `P[q]`. До частки додається байтовий вміст блоку разом з його індексом, що забезпечує правильне впорядкування блоків під час подальшого відновлення зашифрованого вектора.

Частка дилера D формується після поділу зашифрованого вектора на блоки. Для цього виконується побайтове додавання за модулем 2 усіх блоків. У цій реалізації D зберігається як окремий байтовий вектор і використовується лише для обчислення одного відсутнього блоку під час відновлення.

Відновлення використовує індекси блоків, збережені у частках. Під час зчитування доступних часток кожний новий отриманий блок вставляється у позицію, що відповідає його індексу. Одночасно накопичується сума за модулем 2 усіх отриманих блоків без повторень. Якщо після зчитування доступних часток відсутній один блок, його значення визначається операцією $B_{\text{відсут}} = D \oplus X$ і вставляється у відповідну позицію. Після цього блоки об'єднуються за індексами, утворюючи відновлений зашифрований вектор S . Така процедура відновлення наведена на рис. 3.31.

```

RestoreSimplified(P_available, D)
{
    blocks = empty;
    X = 0;

    for each unique received block (j, data):
        blocks[j] = data;
        X = X XOR data;

    if one block is missing:
        blocks[miss] = D XOR X;

    C = JoinBlocks(blocks);

    return C;
}

```

Рисунок 3.31 – Процедура відновлення за спрощеним методом

3.5 Застосування запропонованих (k,n) -схем

Розподілені середовища передавання та зберігання даних передбачають використання множини окремих вузлів, між якими передаються або зберігаються та обробляються частини зображення. На відміну від централізованої моделі, де дані зосереджуються на одному сервері або передаються через один визначений канал, у розподіленому середовищі функції передавання, зберігання та доступу можуть бути розподілені між кількома незалежними або частково незалежними вузлами. Такий підхід застосовується в пірингових мережах, розподілених сховищах, резервних інфраструктурах, а також у системах, де необхідно зменшити залежність від одного вузла або одного маршруту передавання.

У задачах захисту секретного вмісту зображень розподілене середовище може використовуватись не лише як технічний спосіб передавання або зберігання даних, а як частина протоколу безпечного доступу. Якщо повний вектор зображення передається або зберігається як єдиний об'єкт, окремий вузол або канал може отримати весь обсяг даних. Навіть за умови попереднього шифрування це створює концентровану точку спостереження, компрометації або втрати. Тому доцільним є подання зашифрованого вмісту у вигляді множини часток, кожна з

яких має обмежену інформативність і не дозволяє самостійно відновити початкове зображення.

Запропоновані (k,n) -схеми розподілу дозволяють поєднати попереднє шифрування зображення з подальшим розміщенням або передаванням його часток у розподіленому середовищі. У такій моделі зображення спочатку перетворюється у зашифрований вектор, після чого цей вектор поділяється на частки відповідно до заданого правила доступу. Відновлення можливе лише за наявності допустимої множини часток, що визначається параметрами схеми. Це дозволяє уникнути зберігання повного зашифрованого вектора на одному вузлі та забезпечити контрольоване відновлення даних у разі доступності лише частини елементів розподіленої системи.

Далі розглядаються два прикладні сценарії застосування запропонованих схем у таких розподілених середовищах. Перший сценарій пов'язаний з анонімною передачею секретного вмісту зображення через пірингову мережу, де коаліційна (k,n) -схема використовується для розподілу часток між різними вузлами передавання. Другий сценарій стосується резервного копіювання секретного вмісту зображень у контрольованому розподіленому середовищі, де спрощена (k,n) -схема з перекриттям блоків використовується для організації відновлення даних за допустимою кількістю резервних часток.

3.5.1 Протокол анонімної передачі секретного вмісту зображення на основі коаліційної (k,n) -схеми

У розподілених мережевих середовищах передавання або зберігання зображень може виконуватися не лише між двома безпосередньо з'єднаними сторонами, а через множину проміжних вузлів, які беруть участь у пересиланні окремих частин. Для секретного вмісту зображення така модель є доцільною не тільки через можливість рознесення даних між різними елементами мережі, а й через потребу зменшити спостережуваність взаємодії між відправником і кінцевим

отримувачем, забезпечити анонімність сторін. У цьому контексті пірингові мережі можуть бути використані як технічна основа для побудови багатомаршрутної передачі, за якої окремі частини даних проходять через різні проміжні вузли, а встановлення цілісного комунікаційного зв'язку між сторонами ускладнюється.

Задача полягає у розробці протоколу передавання секретного вмісту зображення через пірингову мережу (P2P), завдяки якому взаємодія між відправником і кінцевим отримувачем не зводиться до прямого або сталого маршруту передавання. Для цього процес доставки має бути організований через множину проміжних вузлів і декілька маршрутів, що зменшує спостережуваність повного шляху передавання та ускладнює встановлення належності окремих мережевих потоків до одного сеансу обміну. В такій постановці P2P мережа може використовуватись як середовище, в якому окремі вузли беруть участь у пересиланні даних, але не повинні отримувати повну інформацію про взаємодію між відправником і отримувачем.

Для забезпечення такої передачі даних зображення спочатку шифрується на стороні відправника. Після цього зашифрований вектор розподіляється за узагальненою (k,n) -схемою на індивідуальні частки $P_1, P_2, \dots, P_n$ та частку дилера D . Сформовані частки передаються через вузли пірингової мережі як окремі пакети.

У протоколі беруть участь відправник S , отримувач R , множина вузлів пірингової мережі та дилерська частка D . Множина вузлів може бути представлена як матриця розміром $n \times m$:

$$\left\{ \begin{array}{c} H_{11}, H_{12}, \dots, H_{1n} \\ H_{21}, H_{22}, \dots, H_{2n} \\ \vdots \\ H_{m1}, H_{m2}, \dots, H_{mn} \end{array} \right\},$$

де H_{mn} – n -й вузол m -го шару. Такий запис дозволяє описати передавання часток через кілька послідовних рівнів пірингової мережі, де пакет може переходити від вузла одного шару до вузла наступного шару.

Для кожного вузла H_{mn} задається пара ключів (PK_{mn}, SK_{mn}) , де PK_{mn} – відкритий ключ вузла, SK_{mn} – його приватний ключ. Відкриті ключі проміжних вузлів розміщуються у таблиці T_H доступних вузлів:

$$T_H = \{(ID_{mn}, IP_{mn}, PK_{mn}) \mid H_{mn} \in H\},$$

де ID_{mn} – ідентифікатор вузла, IP_{mn} – його мережева адреса. Кінцевий отримувач також має пару ключів (PK_R, SK_R) . Відкритий ключ PK_R використовується для захисту інформації, призначеної тільки отримувачу, а приватний ключ застосовується ним під час обробки отриманих пакетів.

Для кожної частки P_q , де $q=1, 2, \dots, n$, формується окремий маршрут передавання. Маршрут визначається як послідовність проміжних вузлів, вибраних із різних шарів матриці H , та кінцевого отримувача:

$$L_q = (H_{11}, H_{23}, \dots, H_{mn}, R).$$

Пересилання даних між вузлами може задаватись довільно або псевдовипадково. Тобто пакет, який на першому кроці проходить через вузол H_{11} , на другому кроці може бути переданий не до вузла з тим самим номером, а до іншого вибраного вузла H_{23} . Такий спосіб маршрутизації ускладнює зловмиснику зіставлення вхідних і вихідних потоків між сусідніми шарами та зменшує можливість встановлення повного шляху передавання від відправника до отримувача.

Для передавання частки P_q формується один інкапсульований пакет:

$$PCKT_q = (P_q, X_q^R, X_q^L),$$

де X_q^R – службовий блок для кінцевого отримувача, X_q^L – пошарово захищена маршрутна інформація для проміжних вузлів маршруту.

Службовий блок для кінцевого отримувача містить інформацію, необхідну для коректного використання частки під час відновлення: номер частки q , її позицію

pos_q , індекси блоку або блоків у складі частки I_{dx} , а також частку дилера D . Цей блок шифрується відкритим ключем отримувача:

$$X_q^R = Enc_{PK_R}(q, pos_q, I_{dx}, D).$$

Проміжні вузли не мають доступу до вмісту X_q^R , оскільки не володіють приватним ключем SK_R . Відповідно, вони не знають індекси блоку або блоків у складі частки, не отримують частку дилера та не можуть визначити роль конкретної частки у процедурі відновлення.

Маршрутний блок X_q^L формується покроково відповідно до заданого маршруту L_q . Для кожного проміжного вузла шифрується лише адреса, що вказує наступний проміжний вузол (або кінцевий вузол).

Додатковою властивістю запропонованого протоколу є пороговість відновлення. Оскільки зашифрований вектор попередньо розподіляється за коаліційною (k,n) -схемою, для його відновлення не обов'язково отримувати всі n сформованих часток. Кінцевий отримувач може виконати відновлення після надходження допустимої множини з k часток за умови наявності частки дилера D . Це важливо для пірінгових мереж, де окремі вузли або маршрути можуть бути тимчасово недоступними, а частина пакетів може надходити із затримкою або не бути доставленою. У такому випадку протокол не потребує обов'язкової доставки всіх пакетів, а допускає відновлення за достатньою кількістю коректно отриманих часток.

Отже, отримувач R накопичує частки, що надходять різними маршрутами. Якщо кількість отриманих коректних часток є меншою за k , відновлення неможливе. Після отримання допустимої множини часток отримувач застосовує процедуру відновлення за заданою (k,n) -схемою з використанням дилерської частки D . У результаті відновлюється зашифрований вектор C , після чого виконується розшифрування та зворотнє формування зображення.

На рис. 3.32 наведено приклад схеми протоколу для передачі зашифрованих часток зображення між двома учасниками схеми по заданій кількості вузлів.

Отже, застосування коаліційної (k,n) -схеми для анонімної передачі секретного вмісту зображення дозволяє поєднати попереднє зашифрування даних із їх багатомаршрутним передаванням через пірингову мережу. Це зменшує спостережуваність прямого зв'язку між відправником і кінцевим отримувачем та ускладнює встановлення повного маршруту передавання. Тобто, забезпечується як анонімність зображення так і анонімність користувачів, що обмінюються зображенням.

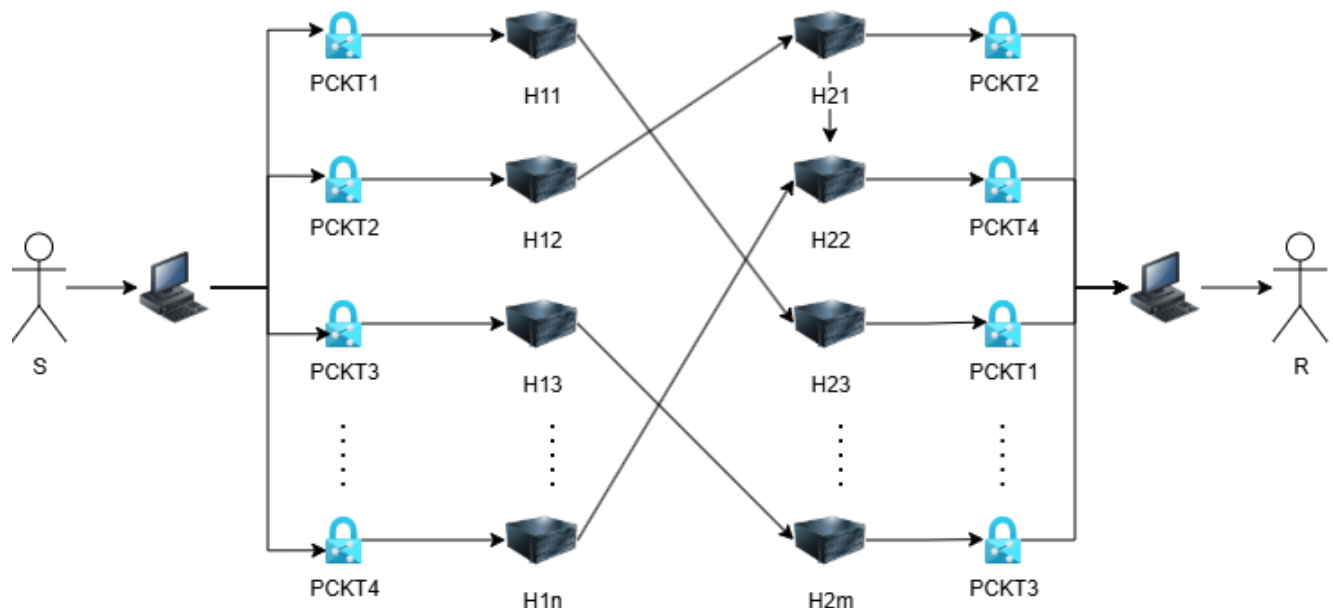


Рисунок 3.32 – Схема протоколу анонімного розподілу часток через проміжні вузли

3.5.2 Протокол застосування спрощеної (k,n) -схеми для зберігання чутливих медичних даних

Зберігання чутливих даних у медичній сфері пов'язане з необхідністю одночасного забезпечення конфіденційності, контрольованого доступу та можливості відновлення інформації у випадку відмови окремих вузлів, втрати частини даних або тимчасової недоступності окремих учасників інфраструктури.

До таких даних можуть належати медичні зображення, результати діагностичних досліджень, фрагменти електронних медичних записів або інші файли, для яких критичними є цілісність, захист від несанкціонованого доступу та збереження можливості подальшого використання. Централізоване зберігання таких файлів у повному вигляді створює ризик компрометації всього вмісту внаслідок доступу до одного вузла або одного сховища. Тому доцільним є застосування такого підходу, за якого файл після криптографічного перетворення не зберігається як єдиний об'єкт, а розподіляється між кількома учасниками або вузлами зберігання.

У межах протоколу, що пропонується, суб'єкт M , який володіє медичним файлом F , виконує зашифрування файла, формування часток за спрощеною (k,n) -схемою та передавання часток учасникам зберігання. Суб'єктом M може бути лікар, медична інформаційна система, діагностичний модуль або інший авторизований компонент, що має право працювати з відповідним файлом. Множина учасників зберігання задається як:

$$\mathbf{P} = \{P_1, P_2, \dots, P_n\},$$

де кожний учасник P_i отримує відповідну частку S_i . Учасниками можуть бути інші лікарі, сервери підрозділів, захищені вузли медичної інформаційної системи або довірені компоненти інфраструктури.

Вхідним об'єктом протоколу є файл медичних даних F , який обробляється як послідовність байтів. Такий спосіб подання дозволяє застосовувати однакову процедуру до медичних зображень різних форматів, архівів досліджень та інших файлів, не прив'язуючи протокол до конкретної структури контейнера.

Відновлення можливе за наявності допустимої кількості часток $P_i \geq k$, для окремих конфігурацій, додатково може існувати необхідність застосування додаткової частки S_D , яка міститься в уповноваженого компонента для відновлення.

У результаті формування захищеного розподіленого набору отримується:

$$\mathbf{A} = \{\mathbf{S}, \mathbf{Q}, S_D, \mathbf{MB}\},$$

де $\mathbf{S} = \{S_1, S_2, \dots, S_n\}$ – множина сформованих часток, $\mathbf{Q} = \{q_1, q_2, \dots, q_n\}$ – відповідає порядковим номерам сформованих часток, S_D – частка, що зберігається у суб'єкта M та використовується під час відновлення, $\mathbf{MB}$ – службові метадані резервного набору. До службових метаданих належать: час T , коли була створення частка, множина I_{SQ} , що задає набори часток, які належать конкретним вузлам зберігання, геш значення $H_S = \{H(S_i)\}$, використовувані для перевірки цілісності часток. Тобто:

$$\mathbf{MB} = \{n, k, T, I_{SQ}, H_S\}.$$

Параметри k і n задаються з урахуванням обмежень спрощеної схеми, розглянутих раніше.

Формування захищеного розподіленого набору починається з вибору файлу F , який підлягає захищеному зберігання. Суб'єкт M виконує зашифрування файлу $\mathbf{C} = \text{Enc}_K(F)$ з використанням секретного ключа K .

Після цього до зашифрованого вектора $\mathbf{C}$ застосовується спрощена (k, n) -схема з перекриттям блоків, у результаті чого формується множина основних часток, частка S_D та службовий блок $\mathbf{MB}$:

$$M : \text{Share}(\text{Enc}_K(F), k, n) \rightarrow \mathbf{S}, \mathbf{Q}, S_D, \mathbf{MB}.$$

Далі суб'єкт M розподіляє основні частки між учасниками зберігання:

$$S_i \rightarrow P_i, \quad i = 1, 2, \dots, n.$$

Для забезпечення можливості відновлення захищеного файлу зображення тільки за участі M передбачено, що M також зберігатиме одну частку розподілених даних P_i . Отже, після надсилання розподілених часток, усі учасники схеми будуть зберігати свою певну частку P_i , а M окрім цієї частки зберігає додаткові службові дані $\mathbf{MB}$ та S_D .

На рис. 3.33 зображена схема розподілу зашифрованого медичного зображення. Тут використані описані вище позначення.

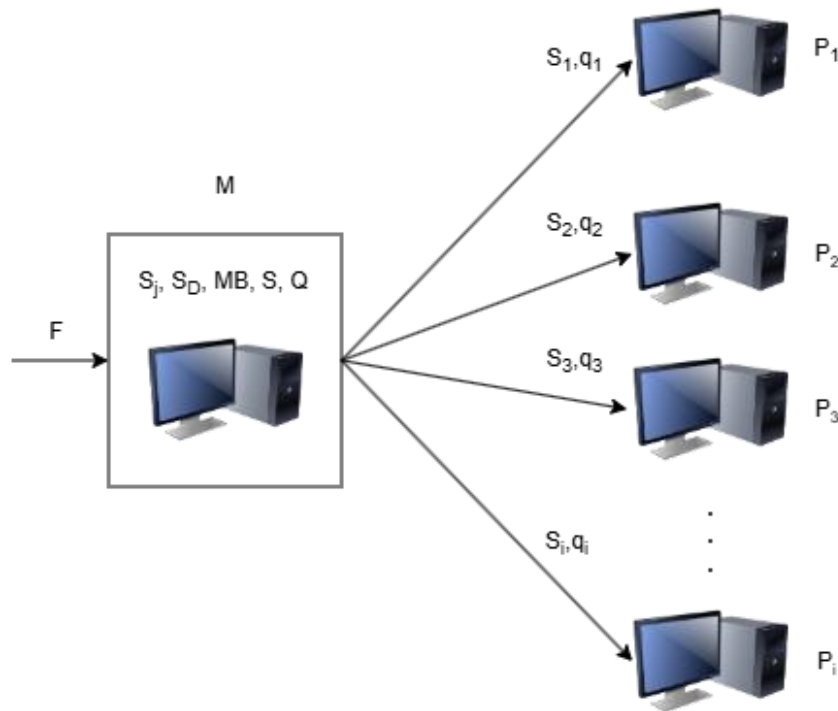


Рисунок 3.33 – Схема розподілу зашифрованого медичного зображення

Відновлення файлу виконується суб'єктом M . Для цього надсилається запитна отримання часток від усіх учасників. Кожен учасник P_i повертає частку разом із її порядковим номером:

$$P_i \rightarrow M : (q_i, S_i).$$

Після отримання часток суб'єкт M виконує їх перевірку за службовими метаданими **MB**. Перевірка передбачає встановлення відповідності між порядковим номером q_i , отриманою часткою S_i та записом у I_{SQ} . Додатково перевіряється цілісність отриманої частки шляхом порівняння її геш-значення з відповідним геш-значенням із множини H_S . Частка використовується для відновлення лише за умови коректної ідентифікації та успішної перевірки цілісності. Локальна частка (q_j, S_j) , що зберігається у M , перевіряється за тим

самим правилом із використанням її порядкового номера та відповідного геш-значення з H_S . Частка S_D перевіряється за геш-значенням H_D . Після цього формується множина часток, що використовуються для відновлення:

$$\mathbf{S}_{nor} = \{S_j, S_{i_1}, S_{i_2}, \dots, S_{i_{q-1}}\}.$$

Для виконання відновлення множина $\mathbf{S}_{nor}$ повинна містити локальну частку S_j та мати кількість часток S_i не меншу за $k-1$. Після формування допустимої та перевіреної множини часток, суб'єкт M виконує відновлення зашифрованого вектора зображення $\mathbf{C}$ з використанням $\mathbf{S}_{nor}$, частки S_D та службових метаданих $\mathbf{MB}$. Далі виконується процедура розшифрування вектора $\mathbf{C}$ та відновлення вхідного зображення F :

$$M : \text{Rec}(\mathbf{S}_{nor}, S_D, \mathbf{MB}) \rightarrow \text{Dec}_K(\mathbf{C}) \rightarrow F.$$

Отже, процедура відновлення передбачає збирання часток від допустимої множини учасників, перевірку їхньої належності та цілісності за службовими метаданими $\mathbf{MB}$, збирання вектора $\mathbf{C}$ та подальше розшифрування. Наявність порядкових номерів q_i дозволяє однозначно ідентифікувати отримані частки, а геш-значення забезпечують виявлення пошкодження або підміни частки до початку відновлення. На рис. 3.34 зображена схема відновлення зашифрованого медичного зображення.

У результаті запропонований протокол задає прикладну модель захищеного розподіленого зберігання чутливого медичного файлу, у якій доступність даних не залежить від одного вузла, а відновлення пов'язується з допустимою множиною учасників та контрольними компонентами, що залишаються у суб'єкта M . Така організація дозволяє зменшити ризик повної компрометації файлу у випадку доступу до окремої частки, оскільки кожний учасник зберігає лише фрагмент розподіленого представлення.

Додаткове використання порядкових номерів часток і контрольних геш-

значень забезпечує перевірку коректності отриманих даних до початку реконструкції.

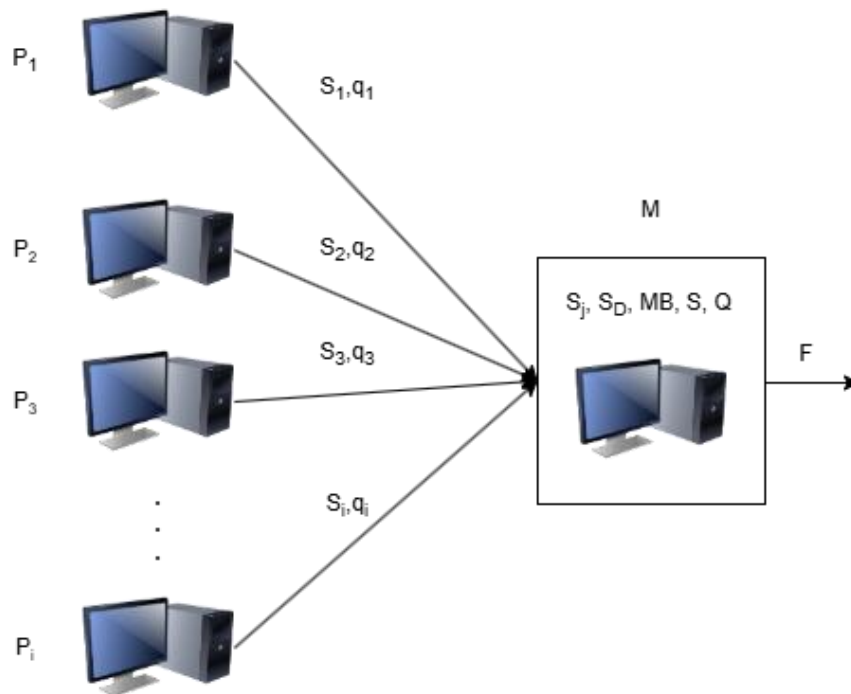


Рисунок 3.34 – Схема відновлення зашифрованого медичного зображення

Це дає змогу виявляти пошкоджені, підмінені або некоректно повернуті частки ще на етапі збирання, тобто до виконання відновлення зашифрованого представлення файлу. У медичному середовищі така властивість є суттєвою, оскільки помилка в частці може призвести не лише до неможливості відновлення, а й до формування некоректного файлу після реконструкції.

3.6 Висновки до третього розділу

У третьому розділі розроблено засоби програмної реалізації запропонованих методів шифрування та розподілу секретного вмісту зображень. Визначено загальну структуру даних і програмних компонентів, що забезпечують послідовну обробку байтового представлення зображення, процес генерації псевдовипадкових чисел, виконання підстановки та перестановки, поділ зашифрованого вектора на частки, а також відновлення початкового зашифрованого вмісту за допустимою

множиною часток. Така структура дозволяє відокремити функціональні частини програмного засобу та забезпечити узгодженість між теоретичним описом методів і їх практичною реалізацією.

Розроблено програмну реалізацію методу шифрування зображення у двох варіантах — SP та PS. Варіант SP реалізує послідовне виконання підстановки та перестановки з формуванням проміжного результату, що дає змогу наочно відокремити два основні етапи криптографічного перетворення. Варіант PS передбачає поєднання підстановки та визначення нової позиції байту в одному проході по вхідному вектору, що зменшує кількість проміжних операцій та потребу в додатковому буфері.

Також у розділі реалізовано програмні засоби для коаліційного та спрощеного методів розподілу секретного вмісту зображення за (k,n) -схемою. Для коаліційного методу визначено порядок поділу зашифрованого вектора на блоки, формування часток відповідно до допустимих коаліцій учасників, збереження індексної інформації та використання допоміжної складової дилера під час відновлення. Для спрощеного методу реалізовано циклічне формування часток із перекриттям блоків, що забезпечує регулярну структуру розподілу та спрощує процедуру програмної реалізації.

Окремо розглянуто практичне застосування запропонованих (k,n) -схем у двох протоколах. Перший протокол призначений для анонімної передачі секретного вмісту зображення у піринговій мережі та передбачає розподіл зашифрованого зображення на частки, передавання їх різними маршрутами через проміжні вузли, використання службових блоків і перевірку цілісності даних. Другий протокол орієнтований на розподілене зберігання чутливих зображень, у якому дані попередньо зашифровуються, розподіляються між вузлами зберігання та можуть бути відновлені лише за наявності допустимої множини часток. Це показує можливість застосування запропонованих методів не лише як окремих

алгоритмічних процедур, а і як складових прикладних систем захищеного передавання, зберігання та відновлення зображень.

Розроблені програмні компоненти забезпечують виконання повного циклу обробки: зашифрування зображення, формування часток, відновлення зашифрованого вектора та подальше розшифрування. Запропоновані протоколи демонструють прикладне використання розроблених (k,n) -схем у задачах анонімної передачі та розподіленого відмовостійкого зберігання зображень. Отримані у цьому розділі результати створюють основу для подальшої експериментальної перевірки коректності, продуктивності та властивостей запропонованих методів у четвертому розділі дисертаційної роботи.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ МЕТОДІВ ШИФРУВАННЯ ТА РОЗПОДІЛУ ЗОБРАЖЕНЬ

У попередніх розділах було розроблено метод шифрування зображень, схеми розподілу секретного вмісту та протоколи їх застосування. У цьому розділі наведено результати дослідження властивостей запропонованих рішень за даними програмної реалізації. Основну увагу приділено характеристикам, що визначають придатність методу до практичного використання: статистичним властивостям зашифрованого представлення, часовим витратам виконання основних процедур, параметрам формування та відновлення часток, а також стійкості протоколу анонімної передачі до типових загроз розподіленого середовища.

Для забезпечення порівняння з відомими методами шифрування та розподілу зображень у частині статистичних експериментів використовується декодоване піксельне подання тестових зображень. Для запропонованого методу таке подання розглядається як послідовність байтових значень, до якої застосовується процедура зашифрування. Конкретні параметри тестових даних, умови вимірювання та конфігурації схем подаються безпосередньо в тих підрозділах, де наведено відповідні результати.

4.1 Дослідження статистичних властивостей методу шифрування

Статистичні властивості методу шифрування досліджуються для зашифрованого представлення зображення, отриманого після виконання операцій підстановки та перестановки. Таке оцінювання характеризує результат першого етапу обробки секретного вмісту зображення до формування часток за схемами розподілу. Основну увагу приділено розподілу байтових значень, ентропії, кореляційним залежностям між сусідніми елементами та бітовим властивостям зашифрованих даних.

4.1.1 Аналіз розподілу значень та ентропії

Однією з основних характеристик зашифрованого зображення є розподіл значень його елементів. Для природних зображень такий розподіл зазвичай є нерівномірним, що зумовлено наявністю однорідних областей, повторюваних кольорів, плавних переходів яскравості та локальних фрагментів зі схожими значеннями пікселів зображення. У гістограмі відкритого зображення це зазвичай проявляється у вигляді виражених максимумів у певних діапазонах значень.

У цьому підрозділі аналіз виконується для низки тестових зображень, які попередньо приведені до декодованого вигляду у форматі пікселів. Таке подання використовується для побудови гістограм, обчислення ентропії, а також для можливості візуально підтвердити коректність та якість шифрування. Для запропонованого методу масив пікселів розглядається як послідовність байтів, до якої застосовується зашифрування. Для кольорових зображень аналізуються значення компонентів відповідного растрового подання, для зображень у відтінках сірого використовується значення яскравості.

Візуальне представлення результату зашифрування тестового зображення у відтінках сірого (baboon) наведено на рисунку 4.1.

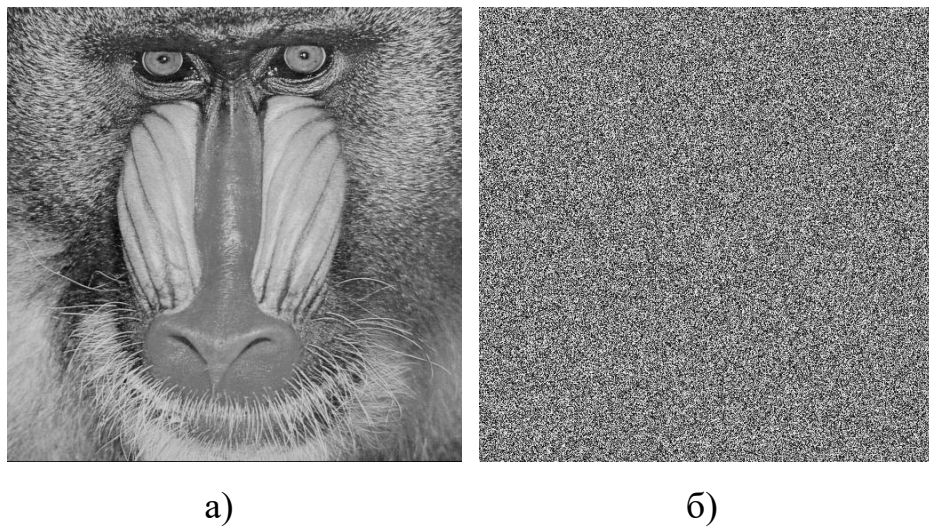


Рисунок 4.1 — Зображення у відтінках сірого:

а) початкове зображення; б) зашифроване представлення

Візуальне представлення результату зашифрування кольорового тестового зображення (peppers) наведено на рисунку 4.2.



а)

б)

Рисунок 4.1 – Кольорові зображення:

а) початкове зображення; б) зашифроване зображення

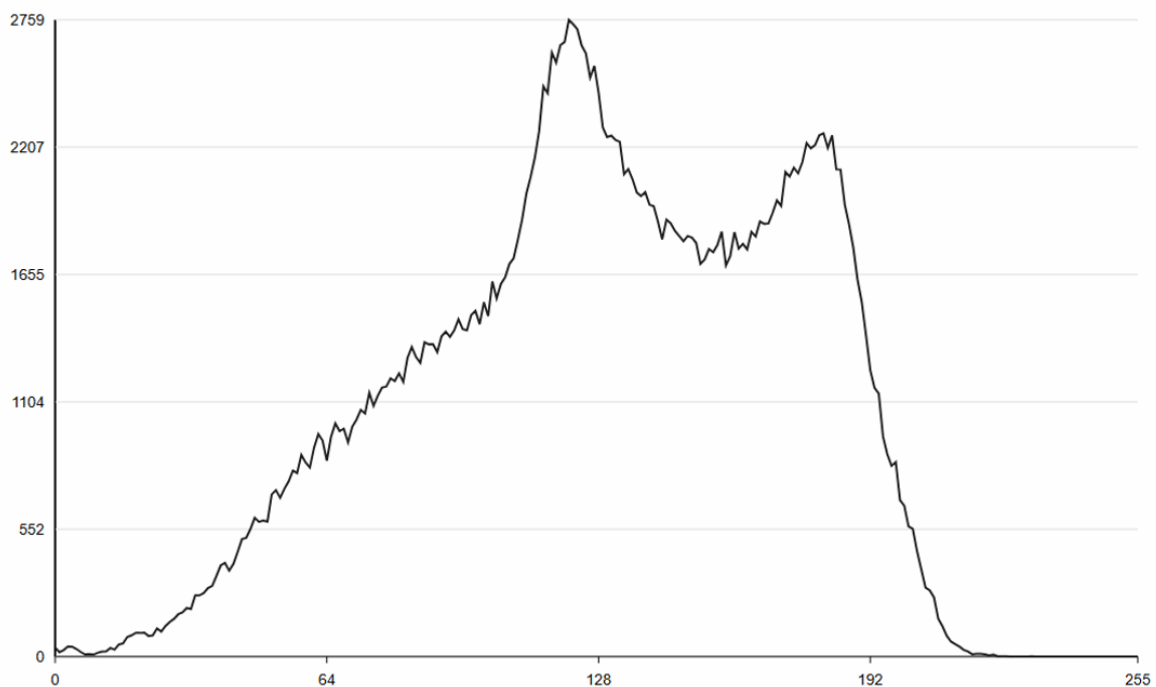
Зашифроване зображення не містить елементів, які б вказували на вміст початкового зображення.

Після зашифрування очікуваною властивістю є вирівнювання частот появи значень пікселів у діапазоні від 0 до 255. У запропонованому методі це формується за рахунок поєднання побайтової підстановки та перестановки елементів. Для перевірки цієї властивості побудовано гістограми відкритих і зашифрованих даних. На рис. 4.2 наведено приклад гістограм для початкового та зашифрованого тестового зображення у відтінках сірого (baboon).

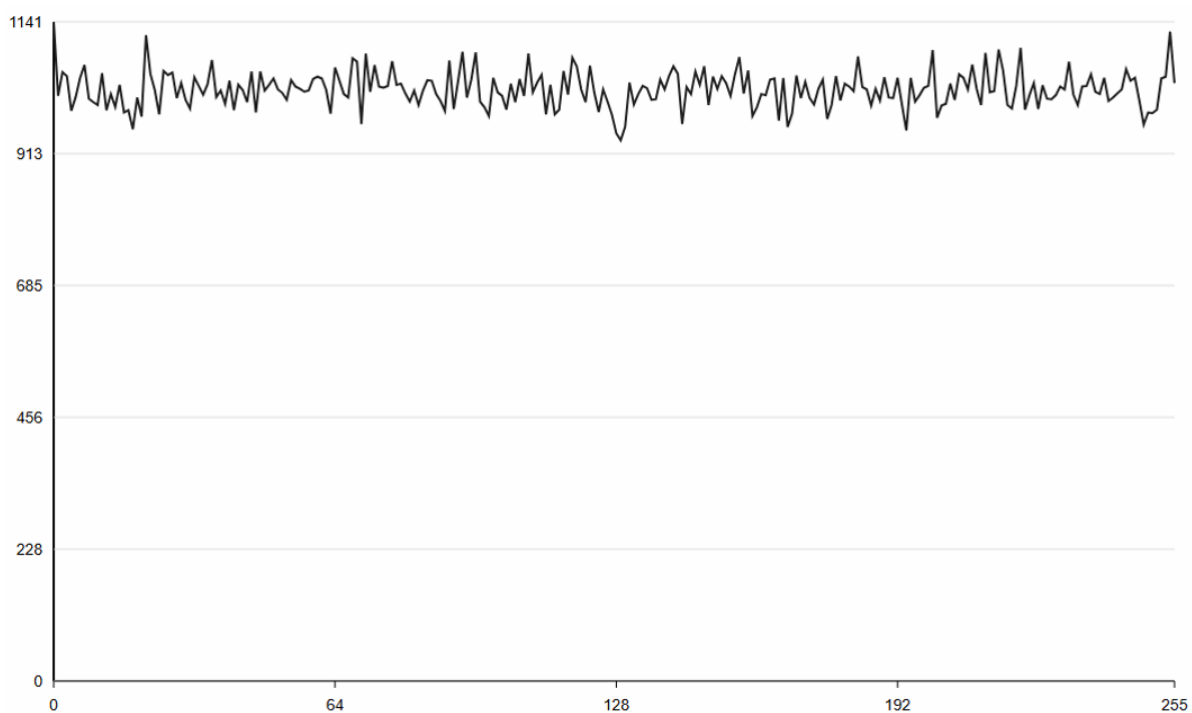
З наведених гістограм видно, що для початкового зображення характерна нерівномірна концентрація значень у певних діапазонах. Після зашифрування розподіл значень стає більш наближеним до рівномірного: частоти появи окремих байтових значень вирівнюються, а виражені максимуми, характерні для відкритого зображення, зникають або істотно зменшуються. Це свідчить про послаблення статистичних ознак, які могли б вказувати на структуру початкового зображення.

На рис. 4.3 наведено приклад гістограм для початкового та зашифрованого

кольорового зображення (peppers).



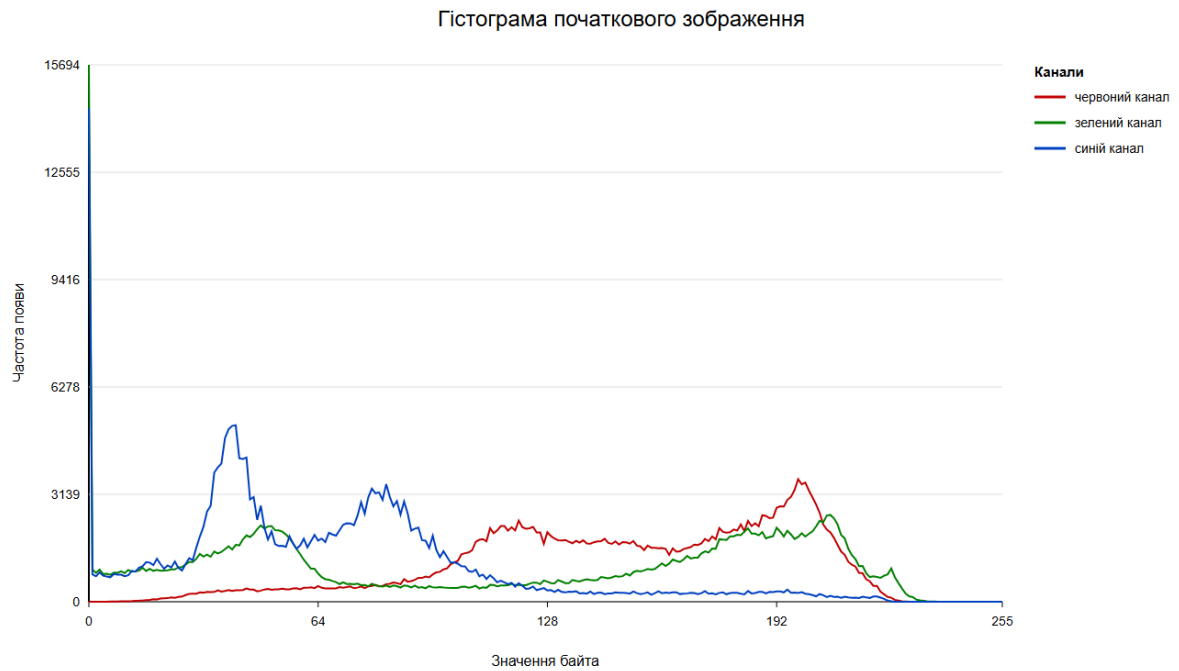
а)



б)

Рисунок 4.2 — Гістограми значень пікселів зображення baboon:

а) початкове зображення; б) зашифроване зображення



а)



б)

Рисунок 4.3 – Гістограми значень пікселів зображення peppers:

а) початкове зображення; б) зашифроване зображення

Кількісною характеристикою рівномірності розподілу є ентропія. Для 8-

бітного подання максимальне значення ентропії становить 8 біт на елемент. Ентропія обчислюється за формулою:

$$H(C) = - \sum_{i=0}^{255} p_i \log_2 p_i,$$

де p_i – ймовірність появи значення i в зашифрованому векторі C . У табл. 4.1 наведено обчислені значення ентропії початкових тестових зображень та після виконання зашифрування.

Отримані результати показують, що після зашифрування ентропія для всіх тестових зображень наближається до граничного значення 8 біт на елемент. Для початкових зображень значення ентропії є нижчими, що відповідає нерівномірному розподілу компонентів природних зображень. Найменше значення ентропії серед відкритих даних отримано для зображення baboon.bmp, що свідчить про вираженішу нерівномірність розподілу значень порівняно з іншими тестовими зразками.

Таблиця 4.1 – Значення ентропії початкових і зашифрованих даних

№	Зображення	Ентропія початкових даних, біт	Ентропія зашифрованих даних, біт
1	baboon.bmp	7.3579	7.9993
2	stone-texture.jpg	7.7557	7.9999
3	peppers.bmp	7.7239	7.9997
4	kyiv-skyline.jpg	7.7563	7.9999

Після зашифрування різниця між значеннями ентропії для різних зображень істотно зменшується: отримані значення перебувають у діапазоні від 7.9993 до 7.9999. Це свідчить про стабільне вирівнювання розподілу байтових значень незалежно від початкового вмісту зображення. Високі значення ентропії не є достатнім доведенням криптографічної стійкості, однак вони підтверджують відсутність вираженої нерівномірності частотного розподілу в зашифрованому представленні.

Отже, аналіз гістограм і значень ентропії показує, що запропонований метод шифрування забезпечує істотне вирівнювання частот появи байтових значень. Це

дозволяє зменшити інформативність частотного аналізу зашифрованого зображення.

4.1.2 Аналіз кореляційних залежностей

Природні зображення характеризуються локальною надлишковістю, оскільки сусідні пікселі або їх компоненти зазвичай мають близькі значення. Це зумовлює високі значення кореляції в горизонтальному, вертикальному та діагональному напрямках. Для зашифрованого представлення бажаною властивістю є зменшення таких залежностей, оскільки збереження високої кореляції може свідчити про наявність залишкової просторової структури відкритого зображення.

Тестові зображення kyiv та stones, що також використані для кореляційного аналізу, наведено на рис. 4.4



а)



б)

Рисунок 4.4 – Тестові зображення: а) kyiv; б) stones

Кореляційний аналіз також виконується для декодованого піксельного подання зображень, оскільки досліджується саме просторова залежність між сусідніми елементами (пікселями) в площині зображення. Горизонтальна кореляція характеризує залежність між сусідніми елементами в одному рядку, вертикальна — між елементами суміжних рядків, діагональна — між елементами, зміщеними одночасно за рядком і стовпцем.

Для пар сусідніх значень x_i та y_i коефіцієнт кореляції обчислюється за формулою:

$$\Psi_{xy} = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^N (y_i - \bar{y})^2}},$$

де N – кількість сусідніх пар пікселів, $\bar{x}$ та $\bar{y}$ – середні значення відповідних вибірок. Результати аналізу кореляції між пікселями початкового та зашифрованого зображень наведено в табл. 4.2.

Для початкових зображень коефіцієнти кореляції мають високі значення, що відповідає природній структурі візуальних даних.

Таблиця 4.2 – Значення кореляції для початкових і зашифрованих зображень

№	Зображення	Горизонтальна кореляція		Вертикальна кореляція		Діагональна кореляція	
		початк.	зашифр.	початк.	зашифр.	початк.	зашифр.
1	baboon.bmp	0.8665	0.0021	0.7586	0.003	0.7261	0.0031
2	stones.jpg	0.9587	0.00011	0.95514	0.00012	0.9276	0.00006
3	peppers.bmp	0.9794	0.00039	0.9802	0.00006	0.9705	0.00127
4	kyiv.jpg	0.9986	0.00016	0.9983	0.00003	0.9976	0.00002

Найбільші значення отримано для зображення kyiv.jpg, де горизонтальна, вертикальна та діагональна кореляція становлять відповідно 0.9986, 0.9983 та 0.9976. Це свідчить про наявність значних однорідних областей і плавних переходів у початковому зображенні. Для baboon.bmp значення кореляції є нижчими, що узгоджується з більш вираженою текстурною структурою цього зображення.

Після зашифрування значення кореляції в усіх напрямках наближаються до нуля. Для зашифрованого зображення kyiv.jpg отримано значення 0.00016, 0.00003 та 0.00002 відповідно для горизонтального, вертикального та діагонального напрямів. Для інших тестових зображень також спостерігається зменшення кореляційних залежностей до значень порядку 10^{-3} – 10^{-5} . З цього видно, що зашифроване представлення не зберігає вираженої лінійної залежності між сусідніми елементами зображення.

Для додаткової візуальної інтерпретації кореляційних залежностей побудовано діаграми розсіювання пар сусідніх елементів. Кожна точка на такій діаграмі відповідає парі значень (x, y) , де x – значення елемента зображення, а y – значення сусіднього елемента у заданому напрямі. Якщо між сусідніми елементами існує сильна залежність, точки концентруються поблизу діагоналі $y = x$ (така залежність зображена на рис. 4.5). Для зашифрованого представлення очікується розсіювання точок по площині без вираженої лінійної структури, що наведено на рис. 4.6.

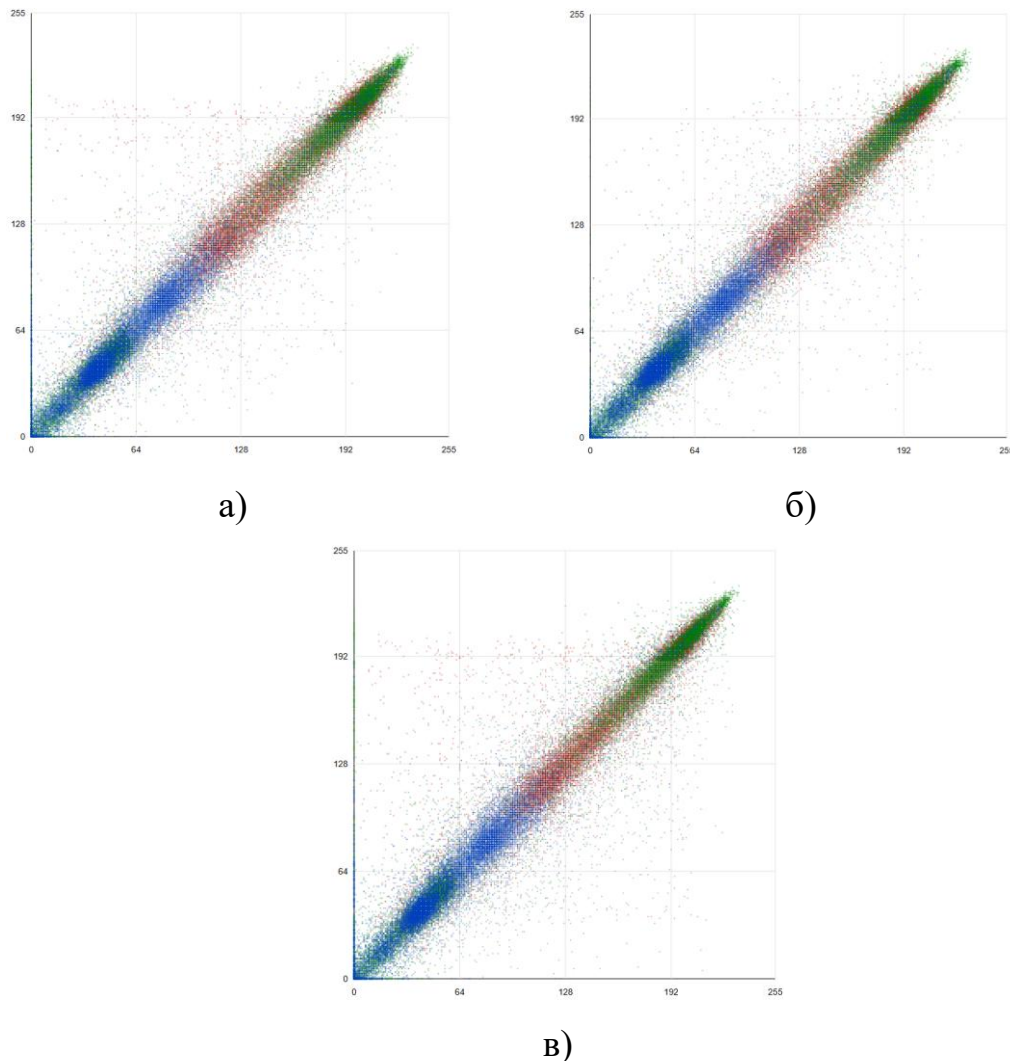


Рисунок 4.5 — Діаграми розсіювання пар сусідніх елементів початкового кольорового зображення (peppers):

а) у горизонтальному; б) вертикальному; в) діагональному напрямках.

зображень. У поєднанні з результатами аналізу розподілу значень та ентропії це свідчить про те, що зашифроване представлення не відтворює частотну та локальну просторову структуру початкового зображення.

4.1.3 Статистичне тестування зашифрованих даних зображень

Для оцінювання бітових властивостей зашифрованого представлення використано набір статистичних тестів NIST SP 800-22 [99]. Тестування спрямоване на виявлення повторюваних значень та патернів у послідовностях бітів, сформованих із результату шифрування, зокрема зміщення частоти нулів та одиниць, надмірної повторюваності блоків, нетипової довжини серій, періодичних структур, відхилень лінійної складності та інших статистичних ознак, які можуть свідчити про нерівномірність бітового подання.

Вхідними даними для тестування є бітові послідовності, сформовані з зашифрованого вектору **C**. Для уніфікації довжини вибірки в кожному випробуванні використовується послідовність довжиною в 1 000 000 біт. До тестової послідовності включаються лише біти зашифрованого зображення без включення службових параметрів.

Тестування виконано для двох типів вхідних зображень: кольорового та зображення у відтінках сірого. Для кожного типу сформовано по 100 бітових послідовностей довжиною 1 000 000 біт. Кількість піддіапазонів у процедурі шифрування є сталою та дорівнювала восьми. Така конфігурація відповідає основному режиму роботи методу, який використовується в подальших експериментальних дослідженнях.

Оцінювання результатів тестування за NIST SP 800-22 виконується для двох показників: частка послідовностей, які пройшли відповідний тест та значенням p для перевірки рівномірності отриманих результатів у серії випробувань. Рівень значущості прийнято рівним $\alpha=0,01$. За умови використання 100 послідовностей очікувана частка проходження тесту має відповідати допустимому інтервалу для

заданого рівня значущості, практично це означає, що кількість успішних проходжень має бути близькою до 99 зі 100 і не опускатися нижче допустимої межі.

Результати проведення тестування за NIST SP 800-22 для кольорового зображення (kuiv) та зображення у відтінках сірого (stones) наведено у табл. 4.3.

Для кожного тесту подано значення P та показник кількості успішно пройдених тестів. Аналіз результатів показує, що сформовані бітові послідовності загалом відповідають статистичним критеріям NIST SP 800-22. Для більшості тестів значення P перевищують пороговий рівень $\alpha=0.01$, а кількість успішно пройдених тестів перебувають на рівні 99/100 або 100/100. Це означає, що переважна більшість послідовностей успішно проходить відповідні статистичні перевірки, а отримані результати не демонструють систематичних відхилень від очікуваних властивостей псевдовипадкової бітової послідовності.

Таблиця 4.3 — Результати статистичного тестування бітових послідовностей

Сіре зображення		Кольорове зображення		Назва тесту
P	Кількість пройдених тестів	P	Кількість пройдених тестів	
0.494	100/100	0.779	100/100	Frequency
0.276	99/100	0.924	100/100	BlockFrequency
0.213	99.5/100	0.375	100/100	CumulativeSums
0.154	98/100	0.972	97/100	Runs
0.276	100/100	0.456	99/100	LongestRun
0.658	100/100	0.739	100/100	Rank
0.456	99/100	0.043	98/100	FFT
0.262	97/100	0.867	97-100/100	Overlapping Template
0.437	99/100	0.304	100/100	Universal
0.720	100/100	0.924	100/100	ApproxEntropy
0.779	100/100	0.494	100/100	LinearComplexity
0.259	99.5/100	0.496	100/100	Serial Median
0.522	99/100	0.525	100/100	NonOverlapping Template Median
0.408	61.5/62	0.286	70/72	RandomExcursions Median
0.410	61/62	0.585	69/72	RandomExcursions Variant Median

4.2 Дослідження продуктивності програмного засобу шифрування

4.2.1 Умови вимірювання та тестові дані

Дослідження продуктивності програмного засобу шифрування, що реалізує запропонований метод шифрування, виконувалося для визначення часових характеристик процедур зашифрування та розшифрування під час обробки зображень різного обсягу. Програмну реалізацію запропонованого методу виконано мовою C++ у середовищі Visual Studio. Для порівняння часових метрик усі реалізації проводились локально за однакових умов та обробляли однакові вхідні дані зображень.

Вхідними даними для вимірювань є значення байтів декодованого до значень пікселів представлення зображень. Для кожного розміру зображення використано один тестовий зразок, оскільки в межах оцінювання продуктивності визначальним фактором є кількість елементів вхідного вектору, адже зміст зображення не впливає на кількість виконуваних операцій у процедурі шифрування.

У часові вимірювання безпосередньо включено виконання процедур зашифрування та розшифрування над отриманим вектором. Операції зчитування початкового файлу, декодування зображення, запису результатів на носій виконуються поза межами вимірюваного інтервалу. Такий порядок дає змогу оцінити швидкість безпосереднього виконання перетворень, які становлять основу методу шифрування. Параметри використаних тестових зображень наведено в табл. 4.4.

Для кожної експериментальної конфігурації виконано 50 запусків. Експериментальні параметри визначаються розміром вхідного зображення, реалізацією методу та напрямом перетворення — зашифруванням або розшифруванням. Як загальний результат обчислюються середні значення часу виконання, отримані за результатами повторних запусків.

Таблиця 4.4 – Параметри тестових зображень

№	Зображення	Розмір, пікселі	Кількість каналів	Обсяг даних, байт
1	baboon.bmp	512×512	1/3	262 144
2	stones.jpg	4016×6016	1/3	24 160 256
3	peppers.tiff	512×512	3/3	786 432
4	kyiv.jpg	5464×3640	3/3	9 789 955

Для характеристики продуктивності використано два показники: середній час виконання та пропускну здатність. Середній час зашифрування позначається як $t_{\text{заши}}$, середній час розшифрування як $t_{\text{розши}}$. Пропускна здатність зашифрування

визначається як відношення обсягу оброблених даних до одиниці часу $R_{\text{заши}} = \frac{S}{t_{\text{заши}}}$

та $R_{\text{розши}} = \frac{S}{t_{\text{розши}}}$ для розшифрування, де S – обсяг вхідного масиву даних у МБ.

Подання результатів у МБ/с дає змогу порівнювати реалізації для зображень різного розміру та оцінювати масштабування часу обробки зі збільшенням обсягу даних.

У кожній реалізації перед виконанням основної процедури зашифрування передбачено стадію ініціалізації. На цій стадії задаються ключові параметри методу, початкові стани генераторів, службові константи, параметри перестановки, підстановки та допоміжні структури, необхідні для подальшої обробки даних. Для запропонованого методу до таких параметрів належать початковий стан РЗЛЗЗ, позиції зворотного зв'язку, кількість піддіапазонів, початкові значення лічильників і коефіцієнти підстановки, що наведені в табл. 4.5. Зазначені параметри залишалися сталими для всіх експериментальних серій цього підрозділу.

Наведені умови забезпечують однакову основу для подальшого порівняння варіантів шифрування SP та PS, а також для порівняння запропонованого методу з відомими методами шифрування зображень за середнім часом виконання та пропускну здатністю.

Таблиця 4.5 – Параметри для процедури шифрування

Параметр	Позначення	Значення
Довжина РЗЛЗЗ	—	64 біти
Кількість бітів для вибору піддіапазону	r	3
Кількість піддіапазонів	2^r	8
Початковий стан РЗЛЗЗ	K_1	0xA1B2C3D4E5F60708
Позиції зворотного зв'язку РЗЛЗЗ	—	$x^{64} + x^{63} + x^{62} + x^{60} + x^{59} + 1$
Початкове значення лічильників	K_2	(0,1,0,1,1,0,0,1)
Напрямок роботи лічильників		
Коефіцієнти для підстановки	a	89
	b	21
	c	112

4.2.2 Порівняння варіантів SP та PS

У межах запропонованого методу шифрування розглядаються два варіанти реалізації послідовності виконання основних перетворень: SP та PS. Варіант SP передбачає послідовне виконання підстановки байтових значень із подальшою перестановкою елементів. Варіант PS передбачає виконання перестановки елементів із подальшою підстановкою байтових значень. Обидві реалізації використовують однакові параметри РЗЛЗЗ, кількість піддіапазонів, початкові значення лічильників і коефіцієнти підстановки, тому отримані часові характеристики відображають вплив порядку виконання перетворень на продуктивність реалізації.

Порівняння SP та PS виконано для тестових зображень різного розміру. Для кожної конфігурації окремо вимірювався час ініціалізації $t_{ініц}$, середній час зашифрування $t_{заш}$, середній час розшифрування $t_{розш}$, а також пропускна здатність зашифрування $R_{заш}$ і розшифрування $R_{розш}$. Час ініціалізації включає підготовку параметрів методу та службових структур, необхідних для виконання подальших перетворень. Пропускна здатність обчислювалася за обсягом вхідного масиву та середнім часом виконання відповідної процедури.

Результати порівняння програмних реалізацій варіантів шифрування SP та PS наведено в таблиці 4.6.

За результатами вимірювань час ініціалізації для обох варіантів є практично сталим і становить близько $1,7 \cdot 10^{-5} - 2,0 \cdot 10^{-5}$ с. Порівняно з часом зашифрування та розшифрування ця величина є незначною, тому основний вплив на продуктивність має безпосереднє виконання підстановки та перестановки над вхідним масивом даних.

Таблиця 4.6 – Результати порівняння програмних реалізацій шифрування за SP та PS варіантами

Обсяг даних, МБ	Метод	$t_{\text{ініц}}$, с	$t_{\text{заши}}$, с	$R_{\text{заши}}$, МБ/с	$t_{\text{розши}}$, с	$R_{\text{розши}}$, МБ/с
0,262	PS	0,000019	0,028	9,3	0,028	9,3
	SP	0,000019	0,048	5,4	0,051	5,1
0,786	PS	0,000018	0,080	9,1	0,080	9,1
	SP	0,000018	0,154	5,1	0,158	4,9
6,56	PS	0,00002	0,705	9,30	0,703	9,33
	SP	0,000017	1,299	5,05	1,343	4,88
23,04	PS	0,00002	2,515	9,16	2,491	9,25
	SP	0,00002	4,541	5,07	4,711	4,9

Для всіх досліджених обсягів даних варіант PS демонструє вищу пропускну здатність порівняно з SP. Під час зашифрування пропускну здатність PS перебуває в межах 9,1-9,3 МБ/с, тоді як для SP вона становить приблизно 5,05-5,4 МБ/с. Аналогічна залежність спостерігається і для розшифрування: для PS отримано значення близько 9,19-9,33 МБ/с, для SP – близько 4,88-5,1 МБ/с.

Зі збільшенням обсягу вхідного масиву час виконання обох варіантів зростає пропорційно до кількості оброблюваних байтів. При цьому пропускну здатність залишається відносно стабільною, що свідчить про переважно лінійний характер масштабування реалізації за обсягом даних. Для варіанта PS час зашифрування зростає від 0,028 с для вектору обсягом 0,262 МБ до 2,515 с для вектору обсягом 23,04 МБ. Для варіанта SP відповідні значення становлять 0,048 с і 4,541 с.

Отримані результати показують, що порядок виконання основних

перетворень впливає на часові характеристики програмної реалізації. Варіант PS забезпечує приблизно у 1,81 рази вищу пропускну здатність порівняно з SP як для зашифрування, так і для розшифрування. З огляду на це, у подальшому зіставленні з відомими методами шифрування зображень використовується варіант PS як продуктивніша реалізація запропонованого методу.

4.2.3 Порівняння з відомими методами шифрування зображень

Для порівняння часових характеристик засобів шифрування, що реалізують запропонований і відомі методи шифрування зображень використовуються адаптовані для досліджень реалізації.

Порівняння виконано з адаптованими програмними реалізаціями відомих методів шифрування зображень, реалізованими мовою C++ [100] і протестовано в однакових умовах. Вхідними даними для кожного методу були однакові підготовлені вектори піксельних компонентів. До вимірювань включено час ініціалізації, час зашифрування та час розшифрування. Операції зчитування початкового файлу, декодування зображення та запису результатів не враховувались. Для кожного методу виконано 30 запусків, після чого визначено середні значення часу та пропускну здатність.

За результатами порівняння варіантів шифрування SP та PS для подальшого зіставлення використано варіант PS, оскільки він забезпечив вищу пропускну здатність зашифрування та розшифрування.

Для оцінювання продуктивності враховано час ініціалізації $t_{iniц}$, час зашифрування $t_{заш}$, час розшифрування $t_{розш}$, пропускну здатність зашифрування $R_{заш}$, пропускну здатність розшифрування $R_{розш}$, а також сумарний час повного циклу $t_{\Sigma} = t_{iniц} + t_{заш} + t_{розш}$. Результати порівняння продуктивності для вхідного вектору обсягом 41,4 МБ наведено в табл. 4.7.

Аналіз результатів показує, що запропонований PS-варіант має найменший

сумарний час повного циклу серед розглянутих реалізацій. Для AES-128-CTR, Dynamic-DNA-chaos та IWT-hyperchaos-Zhong2022 сумарний час виконання перевищує час запропонованого методу у 1,56-1,73 рази. Для Logistic-permutation-XOR час зашифрування та розшифрування є меншим, однак значний час ініціалізації збільшує сумарний час повного циклу у 5,23 разів. Найбільші часові витрати отримано для GF(2⁸)-chaos та CBP-WBPCD, для яких сумарний час перевищує PS-варіант відповідно у 9,04 та 36,85 рази.

Таблиця 4.7 – Порівняльні оцінки продуктивності програмних реалізацій методів шифрування зображень

Метод	$t_{\text{ініц}}$, с	$t_{\text{заши}}$, с	$t_{\text{розши}}$, с	$R_{\text{заши}}$, МБ/с	$R_{\text{розши}}$, МБ/с	t_{Σ} , с	K_{Σ}
AES-128-CTR [92]	0,0000075	4,083	4,088	10,14	10,13	8,171	1,56
Logistic- permutation-XOR [93]	23,336	2,003	2,005	20,66	20,65	27,344	5,23
GF(2 ⁸)-chaos [95]	35,126	6,257	5,872	6,62	7,05	47,255	9,04
Dynamic-DNA- chaos [96]	0,659	4,366	4,028	9,48	10,28	9,053	1,73
CBP-WBPCD [97]	0,062	95,643	96,947	0,434	0,428	192,652	36,85
IWT-hyperchaos- Zhong2022 [98]	2,691	3,294	2,484	12,57	16,67	8,469	1,62
Запропонований метод PS	0,0000035	2,599	2,629	15,93	15,75	5,228	1,00

Отримані результати підтверджують, що запропонований PS-варіант забезпечує кращий сумарний час виконання порівняно з відомими методами шифрування зображень. Це пояснюється малим часом ініціалізації, відсутністю формування повної таблиці перестановки, поєднанням підстановки та перестановки в одному проході, а також використанням математично простих побайтових операцій.

4.3 Дослідження запропонованих схем розподілу секретного вмісту зображення

4.3.1 Аналіз продуктивності запропонованих схем розподілу

У цьому підрозділі наведено результати оцінювання часових характеристик двох варіантів запропонованої (k,n) -схеми розподілу секретного вмісту зображення: коаліційного варіанта та спрощеного варіанта з перекриттям блоків. Етап застосування схеми розподілу охоплює формування часток із зашифрованого вектора $\mathbf{C}$ та подальше відновлення цього вектора за допустимою множиною часток. Час зчитування початкового файлу, декодування зображення та запису результатів у межах цього вимірювання не враховується.

Експеримент виконано для фіксованого зашифрованого вектора $\mathbf{C}$ обсягом 0,25 МБ, який було отримано з кольорового тестового зображення peppers. Для дослідження впливу порогових параметрів, під час тестування пропонується використовувати різні набори значень k та n . Для кожної комбінації (k,n) виконано 30 запусків. У табл. 4.9 наведено середній час формування часток $t_{розп}$, середній час відновлення $t_{відн}$, а також відповідні значення пропускної здатності $R_{розп}$ та $R_{відн}$.

Пропускна здатність формування часток і відновлення визначається як відношення обсягу зашифрованого вектора $\mathbf{C}$ до відповідного часу виконання операції.

Аналіз табл. 4.9 показує, що збільшення параметра n за фіксованого $k = 2$ призводить до зростання часу формування часток для обох реалізацій. Для коаліційної реалізації час $t_{розп}$ збільшується від 0,0012 с для (2,3) до 0,0023 с для (2,5). Для спрощеної реалізації відповідні значення становлять 0,0012с та 0,0021 с. Це впливає зі збільшення кількості сформованих часток та кількості операцій над визначенням відповідності між блоками та частками під час розподілу.

Таблиця 4.9 – Результати тестування запронованих (k,n) -схем розподілу зображення

(k,n)	Спрощена схема				Коаліційна схема			
	$t_{розп}, \text{с}$	$R_{розп}, \text{МБ/с}$	$t_{відн}, \text{с}$	$R_{відн}, \text{МБ/с}$	$t_{розп}, \text{с}$	$R_{розп}, \text{МБ/с}$	$t_{відн}, \text{с}$	$R_{відн}, \text{МБ/с}$
(2,3)	0,00123	202,82	0,0107	23,26	0,00120	207,03	0,0108	23,07
(2,4)	0,0019	130,54	0,0128	19,43	0,0017	146,15	0,0114	21,75
(2,5)	0,0023	107,78	0,0142	17,50	0,0021	116,00	0,0121	20,62
(3,5)	0,0020	119,80	0,0147	16,94	0,0019	131,52	0,0133	18,66

Спрощена реалізація з перекриттям блоків має менший час формування часток для всіх досліджених параметрів (k,n) . Найменша різниця спостерігається для (2,3), де обидві реалізації мають близькі результати за часом розподілу та відновлення. Для інших випадків спрощена реалізація забезпечує вищу пропускну здатність формування часток.

Час відновлення для обох реалізацій є більшим за час формування часток. Для коаліційної реалізації $t_{відн}$ перебуває в межах від 0,0107 до 0,0147 с, для спрощеної реалізації — від 0,0108 до 0,0133 с. За більшістю параметрів спрощена реалізація також має вищу пропускну здатність відновлення. Винятком є (2,3), де значення $R_{відн}$ для обох реалізацій практично збігаються.

Отримані результати показують, що спрощений варіант з перекриттям блоків має кращі часові характеристики для більшості досліджених параметрів (k,n) . Коаліційний варіант є дещо повільнішим, що пов'язано з більш складною організацією допустимих наборів часток. Подальший аналіз обсягу часток і дилерської складової виконується окремо, оскільки ці показники характеризують надлишковість даних, а не швидкість формування та відновлення зашифрованого вектора.

4.3.2 Порівняння обсягу часток запропонованих схем розподілу

Після формування часток важливим показником є обсяг даних, який необхідно зберігати або передавати для подальшого відновлення секрету у вузлах учасників схеми. Для зображень цей показник має практичне значення, оскільки збільшення обсягу часток безпосередньо впливає на вимоги до розподіленого зберігання даних. Для оцінювання обсягу даних, що формується після розподілу зашифрованого вектора C , запропонованих (k,n) -схем порівняно з локальними реалізаціями відомих схем розподілу секрету. Порівняння виконано для параметрів $(k,n) = (3,5)$. Обсяг вхідного зашифрованого вектора в усіх реалізаціях становить $O_C = 1$ МБ.

Порівняння здійснено з такими схемами розподілу секрету: класична схема Шаміра [1], модифікована Ramp схема Шаміра [101], схема Асмута-Блума [31], Блеклі [21] та CFSS схема [102].

Для порівняння використано такі показники: обсяг однієї частки O_{S_i} , сумарний обсяг усіх часток O_Σ , обсяг частки дилера O_D , загальний обсяг даних $O_{заг}$. Для відомих схем частка дилера не використовується, тому $O_{заг} = O_\Sigma$. Для запропонованої схеми повний обсяг враховує індивідуальні частки учасників і частку дилера. Обсяги часток наведено в табл. 4.10.

Таблиця 4.10 – Обсяги часток

Метод	O_{S_i} , МБ	O_Σ , МБ	O_D , МБ	$O_{заг}$, МБ
Коаліційна	0,4	2	0,1	2,1
Спрощена	0,4	2	0,2	2,2
Шаміра	1	5	–	5
Ramp	0,5	2,5	–	2,5
Асмута-Блума	2	10	–	10
Блеклі	2	10	–	10
CFSS	1	5		5

Аналіз табл. 4.10 показує, що для параметрів $(3,5)$ коаліційна та спрощена схеми формують менший обсяг однієї частки порівняно з іншими розглянутими

методами. Обсяг однієї частки для запропонованих схем становить 0,4 МБ, тоді як для Ramp – 0,5 МБ, для Шаміра та CFSS – 1 МБ, а для Асмута-Блума та Блеклі – 2 МБ. Це означає, що в запропонованих схемах окремих учасник зберігає менший фрагмент даних, ніж у порівнюваних методах.

За сумарним обсягом часток коаліційна та спрощена схеми також мають менші значення. Для них $O_{\Sigma} = 2$ МБ, тоді як для Ramp цей показник становить 2,5 МБ, для Шаміра та CFSS – 5 МБ, а для Асмута-Блума та Блеклі – 10 МБ. Отже, за однакового обсягу вхідного вектора $\mathbf{C}$ запропоновані схеми потребують меншого обсягу даних, розміщених у частках учасників.

Частка дилера враховується лише для коаліційної та спрощеної схем. Її обсяг становить 0,1 МБ для коаліційної схеми та 0,2 МБ для спрощеної. З урахуванням цієї частки повний обсяг даних становить відповідно 2,1 МБ і 2,2 МБ. Ці значення залишаються меншими за повний обсяг даних у всіх порівнюваних методах, зокрема меншими за 2,5 МБ для Ramp, який є найближчим за цим показником.

Таким чином, для параметрів (3,5) коаліційна та спрощена схеми забезпечують менший повний обсяг даних порівняно з усіма розглянутими схемами. Порівняно з Ramp, який має найближче значення $O_{\text{заг}} = 2,5$ МБ, повний обсяг коаліційної схеми є меншим приблизно в 1,19 раза, а спрощеної — приблизно в 1,14 раза. Порівняно зі схемами Шаміра та CFSS повний обсяг коаліційної схеми є меншим приблизно в 2,38 раза, а спрощеної — в 2,27 раза. Порівняно зі схемами Асмута-Блума та Блеклі з обсягом 10 МБ перевага становить приблизно 4,76 раза для коаліційної схеми та 4,55 раза для спрощеної.

4.3.3 Порівняння продуктивності з відомими схемами розподілу секрету

Для порівняння часових характеристик використано аналогічні схеми розподілу секрету, що розглядалися при оцінюванні обсягу часток. Порівняння виконано для параметрів $(k,n)=(3,5)$ та зашифрованого вектора $\mathbf{C}$ обсягом 1 МБ. Для кожної реалізації виконано 30 запусків. У табл. 4.11 наведено середній час

формування часток $t_{розн}$, середній час відновлення $t_{відн}$, а також відповідні значення пропускну́ї спроможності $R_{розн}$ та $R_{відн}$. При цьому для запропонованих схем розподілу до часу формування та відновлення часток включено час процедур зашифрування та розшифрування відповідно. Для визначеного обсягу вектора цей час становить $t_{заши} = 0,0569$ с для зашифрування та $t_{розши} = 0,0568$ с для розшифрування.

За результатами порівняння коаліційна та спрощена схеми мають найменші часові витрати на формування часток розподілу та відновлення зашифрованого вектора. Найближчим до них за часом формування серед порівнюваних методів є Ramp, для якого $t_{розн} = 0,538$ с.

Таблиця 4.11 – Оцінки продуктивності схем розподілу секрету

Варіант схеми	$t_{розн}$, с	$R_{розн}$, МБ/с	$t_{відн}$, с	$R_{відн}$, МБ/с
Коаліційна	0,0597	16,75	0,1096	9,13
Спрощена	0,0595	16,82	0,1026	9,74
Шаміра	0,993	1,01	3,649	0,27
Ramp	0,538	1,86	5,698	0,18
Асмута-Блума	1,246	0,8	4,906	0,20
Блеклі	0,647	1,54	11,805	0,08
CFSS	0,718	1,39	6,283	0,16

Відносно цього значення час формування часток у коаліційній схемі є меншим приблизно у 9,01 раза, а у спрощеній — приблизно у 9,04 раза. Це показує, що різниця є наслідком принципово простішої процедури формування часток, яка не потребує поліноміального інтерполювання, використання обчислень в межах CRT або розв’язання системи рівнянь в геометричній системі.

Для процедури відновлення найближчим серед порівнюваних методів є схема Шаміра з часом 3,649 с. Коаліційна схема виконує відновлення приблизно у 33,3 раза швидше, а спрощена – приблизно у 35,6 раза швидше. Менша різниця порівняно з етапом формування пояснюється тим, що відновлення в запропонованих схемах передбачає збирання повної структури зашифрованого

вектора з допустимого набору часток і частки дилера, однак ця процедура все одно залишається істотно простішою за відновлення в розглянутих класичних схемах.

За пропускною здатністю отримано аналогічну залежність. Під час формування часток коаліційна схема забезпечує 16,75 МБ/с, а спрощена – 16,82 МБ/с, тоді як найбільше значення серед порівнюваних методів становить 1,86 МБ/с для Ramp. Під час відновлення пропускна здатність коаліційної схеми становить 9,13 МБ/с, спрощеної – 9,74 МБ/с, тоді як найкращий результат серед порівнюваних методів дорівнює 0,27 МБ/с для схеми Шаміра.

Таким чином, для параметрів $k = 3$ та $n = 5$ запропоновані схеми мають перевагу за обома часовими показниками. Спрощена схема є швидшою за коаліційну приблизно у 1,004 раза під час формування часток і приблизно у 1,07 раза під час відновлення. У поєднанні з результатами підрозділу 4.3.2 це означає, що запропоновані схеми забезпечують не лише менший повний обсяг даних, а й істотно менший час виконання процедур розподілу та відновлення порівняно з розглянутими методами.

4.5 Аналіз стійкості запропонованого протоколу анонімної передачі зображення

4.5.1 Методика аналізу стійкості протоколу анонімної передачі зображення

Аналіз запропонованого протоколу анонімної передачі зображення через пірингову мережу виконується для визначення того, як властивості шифрування, розподілу, контролю цілісності та багатомаршрутного передавання впливають на стійкість передавання часток та збереження конфіденційності безпосередніх учасників. У цьому підрозділі стійкість розглядається як сукупність характеристик, що описують захист вмісту, можливість виявлення модифікованих часток, збереження умов відновлення за допустимою множиною даних та анонімізацію взаємодії між відправником і отримувачем.

У дослідженнях анонімних комунікацій і аналізу трафіку анонімність не розглядається як одна неподільна властивість [103]. Зазвичай окремо виділяють анонімність учасника, незв'язуваність подій, неспостережуваність факту взаємодії та приховування відповідності між повідомленням і суб'єктами комунікації. Ці поняття розмежовуються як окремі категорії, що дозволяє аналізувати не лише зміст переданих даних, а й інформацію, яка може бути отримана зі структури взаємодії. Для протоколу передавання часток таке розмежування є доцільним, оскільки зашифрування захищає зміст зображення, тоді як пірингова маршрутизація, інкапсуляція службових даних і рознесення часток різними маршрутами впливають на спостережуваність передавання.

Базова ідея анонімної передачі через проміжні вузли походить із роботи [104], у якій запропоновано приховувати відповідність між відправниками, отримувачами та повідомленнями за рахунок криптографічної обробки й пересилання через проміжні елементи мережі. Для даного дослідження ця ідея використовується як методична основа для аналізу того, наскільки окремий проміжний вузол або стороннє спостереження може встановити зв'язок між відправником, отримувачем і переданою часткою. При цьому об'єктом аналізу є не одиничне повідомлення, а множина часток зашифрованого зображення, що передаються в межах одного прикладного процесу.

Окреме значення має приховування маршрутної інформації. У роботах [105, 106] розглянуто підхід, за якого проміжний вузол отримує тільки ту інформацію, яка потрібна для локального пересилання, без володіння повним маршрутом. Для запропонованого в підрозділі 3.5.2 протоколу ця властивість використовується під час оцінювання обсягу службової інформації, доступної проміжному вузлу пірингової мережі. Якщо вузол має лише локальні дані про пересилання, його компрометація не повинна автоматично давати повний маршрут, кінцевого отримувача або структуру відновлення секретного вмісту.

Пірингова мережа [107] в цьому контексті розглядається як розподілена множина вузлів, через які можуть проходити пакети з частками. Її використання дозволяє формувати маршрути передавання без прямого каналу між відправником і отримувачем та розносити частки між різними вузлами. У роботах [108, 109] пірингові підходи розглядаються як один із напрямів побудови анонімних комунікаційних систем поряд із каскадними моделями змішування. Для цієї роботи релевантним є саме факт використання пірингової топології як основи для маршрутизації часток і зменшення інформативності окремого вузла щодо всього процесу передавання.

Водночас оцінка анонімізації не може обмежуватися лише фактом використання проміжних вузлів. У роботі [110] з аналізу трафіку показано, що зловмисник може отримувати корисну інформацію без розшифрування вмісту, використовуючи час передавання, розмір пакетів, повторюваність маршрутів, напрям руху та інші метадані. Отже, для протоколу передавання часток необхідно оцінювати не тільки можливість отримання відкритого зображення, а й можливість зв'язування окремих пакетів, маршрутних фрагментів, відправника й отримувача.

З урахуванням зазначених підходів аналіз виконується за чотирма основними групами властивостей.

Перша група стосується конфіденційності вмісту. Тут аналізується, чи може нападник отримати відкрите зображення або відновити зашифрований вектор за наявності окремих часток, неповної множини часток або часткового доступу до переданих пакетів. Для цієї групи суттєвими є результати експериментальної перевірки шифрування, розподілу та відновлення.

Друга група стосується цілісності часток. У цьому випадку оцінюється можливість модифікації, підміни або пошкодження частки під час передавання. Основним критерієм є здатність протоколу виявити некоректну частку за геш-значенням до етапу відновлення.

Третя група стосується доступності відновлення. Тут аналізується вплив втрати, затримки або блокування частини пакетів на можливість подальшого відновлення секретного вмісту. Для цієї групи важливим є співвідношення між кількістю сформованих часток, пороговою умовою відновлення та кількістю коректно отриманих часток.

Четверта група стосується анонімізації передавання. Вона охоплює анонімність відправника та отримувача, приховування маршрутної інформації та незв'язність часток. Приховування маршрутної інформації визначає обсяг даних про шлях передавання, доступних окремому вузлу або при сторонньому спостереженні. Незв'язність часток характеризує складність встановлення того, що кілька пакетів належать до одного процесу передавання.

Таким чином, методика оцінювання поєднує змістову та мережеву складові. Змістова складова пов'язана із зашифрованим вектором, частками, контролем цілісності з використанням геш-значень і умовами відновлення. Мережева складова пов'язана з маршрутами, проміжними вузлами, службовою інформацією та спостережуваними параметрами передавання. Такий поділ дозволяє послідовно перейти до моделі зловмисника та провести аналіз окремих груп атак: перехоплення часток, модифікацію пакетів, втрату частини переданих даних, компрометацію проміжних вузлів, аналіз трафіку та Sybil-атаки в піринговій мережі [111].

Отже, у межах подальшого аналізу стійкість протоколу розглядається як сукупність криптографічних, структурних і мережевих властивостей. Криптографічна складова обмежує можливість отримання відкритого вмісту з окремих часток. Структурна складова визначає умови відновлення за допустимою множиною часток. Мережева складова визначає рівень анонімізації передавання через пірингову мережу, обсяг інформації, доступної проміжним вузлам, і стійкість до спостереження за маршрутами.

4.5.2 Модель зловмисника

Аналіз стійкості передавання часток та анонізації протоколу потребує визначення можливостей зловмисника щодо ідентифікації організації мережі, передачі пакетів, структури проміжних вузлів і часток. Для протоколу анонімної передачі зображення через пірингову мережу суттєвими є три групи можливостей: спостереження за мережевими подіями, активний вплив на пакети та контроль частини проміжних вузлів. Таке подання узгоджується з підходами, що використовуються в дослідженнях анонімних комунікацій та аналізу трафіку, де зловмисник характеризується не лише доступом до повідомлення, а й можливістю аналізувати час, обсяг, напрям і повторюваність передавання [112].

У праці з аналізу трафіку [113] пасивний зловмисник зазвичай описується як суб'єкт, що не змінює передані дані, проте може фіксувати параметри комунікації: час появи пакетів, обсяг переданих даних, видимі вузли, напрям руху та повторюваність маршрутних шаблонів. Активний зловмисник має розширені можливості: він може затримувати, видаляти, дублювати або змінювати пакети. Для протоколу передавання часток таке розмежування є важливим, оскільки пасивні дії пов'язані переважно з анонімізацією та зв'язуванням пакетів, тоді як активні дії впливають на цілісність часток, доступність передавання і коректність подальшого відновлення.

Для мереж із проміжними вузлами додатково враховується розташування зловмисника у маршруті передачі. Якщо зловмисник контролює окремий проміжний вузол, він може отримати локальну службову інформацію, необхідну для пересилання пакета. У підходах до приховування маршрутної інформації [114] проміжний вузол оперує лише тими даними, які потрібні для локальної передачі, що обмежує його знання про повний маршрут. Для розглянутого протоколу ця властивість використовується під час аналізу того, чи може один проміжний вузол встановити кінцевого отримувача, повний шлях передавання або роль конкретної частки у процедурі відновлення.

Пірингова мережа визначає окрему складову аналізу, оскільки маршрути передавання часток проходять через множину вузлів, що можуть належати різним суб'єктам. У літературі [115] пірингові моделі розглядаються як один із підходів до побудови анонімних комунікаційних систем, де стійкість залежить від вибору проміжних вузлів, топології маршрутів і частки скомпрометованих учасників мережі. Для протоколу передавання часток це означає, що зломисник може намагатися не тільки перехопити окремий пакет, а й збільшити свою присутність у маршрутах, контролюючи частину вузлів пірингової мережі.

Для пірингових мереж характерною є загроза Sybil-атак. Її сутність полягає у створенні одним реальним суб'єктом великої кількості логічних ідентичностей або вузлів, що дозволяє отримати непропорційно великий вплив на розподілену систему. Оскільки в протоколі передавання часток використовуються маршрути через вузли пірингової мережі, Sybil-атака включається до моделі як окремий вектор мережевого впливу. У цьому випадку її наслідком є підвищення ймовірності потрапляння контрольованих вузлів до маршрутів передавання, накопичення маршрутних спостережень і збільшення шансів перехоплення окремих часток.

Підсумовуючи вище сказане, у табл. 4.12 описано основні моделі зломисника. Дії зломисників можуть бути спрямовані на мережевий, маршрутний і структурний рівні протоколу. Така модель зломисника забезпечує послідовний перехід до оцінки конкретних груп атак.

Зломисники з мережевою видимістю спостерігають за маршрутами передавання пакетів і намагаються пов'язати їх з конкретними учасниками. Активний мережевий зломисник оцінює вплив модифікації, затримки, блокування та повторного використання часток. Скомпрометовані проміжні вузли дають змогу оцінити, який обсяг службової інформації розкривається під час пересилання. Зломисник із недостатньою множиною часток перевіряє чи може неповний набір даних привести до відновлення зашифрованого вектора.

Таблиця 4.12 – Моделі зловмисника

Клас зловмисника	Опис зловмисника	Мета атаки	Доступ і можливості
A1	Зловмисник з локальною видимістю мережевих подій	Встановлення зв'язку між окремими пакетами або напрямом передавання	Фіксує окремий канал, час передавання, розмір пакета та видимі адреси вузлів
A2	Зловмисник із розширеною видимістю мережевих подій	Кореляція пакетів за часом та обсягом, встановлення залежності між кількома частками	Має доступ до кількох каналів або кількох фрагментів маршрутів
A3	Активний мережевий зловмисник	Порушення цілісності часток, повторне використання пакетів, зрив відновлення	Затримує, блокує, дублює або змінює пакети
A4	Скомпрометований проміжний вузол	Встановлення попереднього або наступного вузла, часткове трасування маршруту	Отримує локальну службову інформацію, потрібну для пересилання пакета
A5	Коаліція скомпрометованих вузлів	Відновлення фрагментів маршруту, зіставлення кількох пакетів або часток	Об'єднує спостереження кількох проміжних вузлів
A6	Sybil-зловмисник	Збільшення ймовірності входження контрольованих вузлів до маршрутів	Створює або контролює значну кількість логічних вузлів у піринговій мережі
A7	Зловмисник із недостатньою множиною часток	Спроба відновити зашифрований вектор або отримати інформацію про зображення	Має одну або кілька часток без достатньої множини для відновлення
A8	Зловмисник, що підмінює частки	Внесення пошкоджених даних у процедуру відновлення	Формує змінену або некоректну частку

Отже, модель зловмисника для цього підрозділу охоплює ті можливості, які безпосередньо впливають на стійкість передавання часток: доступ до мережевих метаданих, контроль окремих вузлів, активний вплив на пакети, накопичення часток і спроби підміни даних. Це дозволяє надалі окремо оцінити атаки на вміст,

атаки на цілісність і доступність, а також атаки на анонімізацію передавання через пірингову мережу.

4.5.3 Оцінювання сценаріїв атаки на анонімність при передачі часток

Основною характеристикою, що оцінюється для протоколу передавання часток через пірингову мережу, є забезпечення анонімної взаємодії між відправником S і кінцевим отримувачем R . У цьому контексті зловмисник розглядається передусім як суб'єкт, що намагається встановити зв'язок між учасниками передавання, визначити маршрут проходження пакетів, пов'язати кілька часток з одним процесом передавання або зменшити множину можливих відправників і отримувачів. Властивості зашифрування, коаліційного розподілу, геш-значення та ключової інкапсуляції службових даних у цьому підрозділі враховуються як чинники, що знижують інформативність окремого пакету, окремої частки та окремого проміжного вузла. Тут використовуються позначення, що введені у підрозділі 3.5.1. Крім того, розглядаються як окремі класи зловмисників так і кілька класів одночасно, які характеризуються не сукупністю зловмисників, а сукупністю їх можливостей. Тобто як один зловмисник з такою сукупністю можливостей.

Сценарій 1. Локальне спостереження за окремим фрагментом маршруту.

У цьому сценарії розглядається зловмисник $A1$, який має доступ до окремого каналу або окремої ділянки маршруту проходження пакета $PCKT_q$. Йому можуть бути доступні факт проходження пакета, час передавання, орієнтовний розмір, напрям руху та видимі адреси суміжних вузлів. Такий доступ характеризує лише локальну мережеву подію і не розкриває повний маршрут L_q .

Зловмиснику $A1$ недоступні параметри, що визначають повну роль пакета в протоколі. Він не має SK_R , тому не може розкрити X_q^R , відповідно, йому недоступні D , I_{dx} , додаткові параметри частки та службові дані, необхідні для їх використання під час відновлення. Також локальне спостереження не дає повної

множини часток і не дозволяє встановити, які інші пакети належать до того самого передавання.

Можливий результат атаки обмежується встановленням факту проходження $PCKT_q$ через спостережувану ділянку. На підставі такого доступу зломисник може сформулювати припущення про напрям пересилання, однак не може встановити відповідність між S , R і часткою P_q . Обмеження атаки забезпечується тим, що окремий пакет не містить повного секретного вмісту, службовий блок отримувача захищений ключовою інкапсуляцією, а локальне мережеве спостереження не охоплює повну структуру передавання.

Сценарій 2. Компрометація одного проміжного вузла.

У цьому сценарії розглядається зломисник A_4 , який контролює один проміжний вузол H_{mn} на маршруті передавання частки. Йому доступний факт проходження пакета $PCKT_q$ через цей вузол, час його обробки, локальний напрям пересилання та частина маршрутного блока X_q^L , необхідна для визначення наступного вузла маршруту L_q .

Отриманий доступ дозволяє зломиснику встановити лише локальний фрагмент маршруту. Він може визначити, що пакет проходив через контрольований вузол, і зафіксувати суміжний перехід, однак не отримує повного маршруту від S до R . Також йому недоступний службовий блок отримувача X_q^R , оскільки його розкриття потребує секретного ключа SK_R . Відповідно, зломисник має недостатньо параметрів для можливості встановлення ролі частки P_q у процедурі відновлення. Отже, компрометація одного проміжного вузла не забезпечує повної деанонімізації передавання.

Сценарій 3. Компрометація кількох проміжних вузлів.

У цьому сценарії розглядається зломисник A_5 , який контролює кілька

проміжних вузлів пірингової мережі. Йому можуть бути доступні локальні фрагменти маршрутів L_q , час проходження пакетів через контрольовані вузли, суміжні переходи та відповідні частини маршрутних блоків X_q^L . Порівняно з одним скомпрометованим вузлом, такий доступ дає змогу зіставляти кілька мережевих подій між собою.

Основний ризик полягає у частковій маршрутній кореляції: зловмисник може порівнювати часові ознаки, напрямки пересилання, повторення проміжних вузлів і намагатися пов'язати кілька $PCKT_q$ з одним процесом передавання. Водночас без SK_R він не може розкрити X_q^R , тому не отримує службові параметри для відновлення. Наявність кількох маршрутних фрагментів також не означає доступу до всієї множини часток або підтвердженої належності цих часток до одного процесу відновлення.

Отже, компрометація кількох проміжних вузлів підвищує можливість кореляції маршрутів, але не створює достатніх умов для повної деанонімізації $S \rightarrow R$. Для цього зловмиснику необхідно не лише зіставити маршрутні фрагменти, а й отримати службовий контекст, який у протоколі винесено за межі доступу проміжних вузлів.

Сценарій 4. Кореляція пакетів на основі аналізу часу передавання та обсягу даних.

У цьому сценарії розглядаються зловмисники A1, A2 та частково A5, які не обов'язково мають доступ до змісту пакету $PCKT_q$, але можуть спостерігати його час передавання та обсяг пакетів. До доступних ознак належать час появи пакету t_q , обсяг $PCKT_q$, напрям пересилання, повторюваність маршрутних ділянок і наявність групи пакетів у близькому часовому інтервалі. Метою такої атаки є не розкриття частки P_q , а встановлення того, що кілька мережевих подій можуть належати до одного процесу передавання.

Основний ризик полягає у спробі зіставити декілька $PCKT_q$ за часовими інтервалами, розміром або подібністю маршрутних фрагментів. Проте такі ознаки не розкривають службовий контекст відновлення і не підтверджують роль конкретної частки у формуванні C . Навіть якщо зловмисник припускає, що кілька пакетів пов'язані між собою, він переходить лише до ймовірнісного зв'язування мережевих подій, а не до встановлення повної структури передавання.

Отже, така кореляція є релевантною саме для анонімізації передавання, оскільки може зменшувати невизначеність щодо належності пакетів до одного процесу. Її результатом є ймовірнісне зв'язування мережевих подій, а не пряме встановлення залежності між S та R , або отримання секретного вмісту. Обмеження атаки забезпечується відсутністю відкритих службових ознак у пакетах, ключовою інкапсуляцією X_q^R і можливістю передавання часток різними маршрутами.

Сценарій 5. Зв'язування кількох часток з одним процесом передавання.

У цьому сценарії розглядаються можливості зловмисників $A2$, $A4$ та $A5$, які намагаються визначити, що кілька пакетів $PCKT_q$ належать до одного процесу передавання секретного вмісту зображення. Для цього можуть використовуватися часові ознаки, розміри пакетів, повторення проміжних вузлів H_{mn} , суміжні маршрутні фрагменти та локальні частини X_q^L , доступні контролюваному вузлу. На відміну від кореляції за часом та обсягом окремих подій, тут метою є групування декількох часток як елементів одного передавання.

Навіть у разі часткового зв'язування кількох пакетів зловмисник не отримує повного службового контексту відновлення. Без розкриття X_q^R спостереження за кількома $PCKT_q$ не дає підтвердженого порядку використання часток, не гарантує наявності допустимої множини для відновлення C і не дає параметрів розшифрування. Тому результат атаки залишається на рівні групування пакетів, а не на рівні відновлення секретного вмісту.

Отже, зв'язування кількох часток послаблює анонімізацію, оскільки дозволяє перейти від аналізу окремого пакета до аналізу групи пакетів. Водночас такий результат не є достатнім для відновлення секретного вмісту або однозначного встановлення пари $S \rightarrow R$. У протоколі цей сценарій обмежується рознесенням часток різними маршрутами, локальністю маршрутної інформації в X_q^L і недоступністю службового блока отримувача X_q^R для проміжних вузлів.

Сценарій 6. Встановлення зв'язку між відправником і отримувачем.

У цьому сценарії розглядаються можливості зломисників A2 та A5, які мають розширену видимість мережевих подій або контролюють кілька проміжних вузлів. Їхня мета полягає у встановленні відповідності між відправником S і кінцевим отримувачем R на основі доступних маршрутних та часових ознак. Для цього можуть використовуватися дані про вихідні пакети від S , фрагменти маршруту L_q , час проходження $PCKT_q$ через контрольовані вузли, напрям пересилання та події надходження пакетів до отримувача.

Основний ризик полягає у зіставленні різних фрагментів передавання в один ланцюг $S \rightarrow \dots \rightarrow R$. Такий сценарій є складнішим за зв'язування кількох часток, оскільки зломисник має не лише припустити належність кількох $PCKT_q$ до одного процесу, а й пов'язати цей процес із конкретним відправником і конкретним отримувачем. Без доступу до повного маршруту, службового контексту відновлення та всієї множини часток така відповідність не встановлюється безпосередньо, а може формуватися лише як результат кореляції кількох мережевих ознак.

Отже, встановлення зв'язку $S \rightarrow R$ є основним сценарієм деанонімізації, але його успішність потребує значно ширшого доступу, ніж перехоплення одного пакета або компрометація одного проміжного вузла. Запропонований протокол обмежує цей сценарій за рахунок передавання через проміжні вузли пірингової

мережі, рознесення часток різними маршрутами, локальності маршрутної інформації в X_q^L та захисту службового блока отримувача X_q^R . У результаті окрема точка спостереження не дає повної відповідності між S, R і переданими частками.

Сценарій 7. Встановлення кінцевого отримувача за службовими даними.

У цьому сценарії розглядаються можливості зловмисників A4 та A5, які мають доступ до одного або кількох проміжних вузлів і намагаються визначити кінцевого отримувача R не через загальний аналіз трафіку, а через службову інформацію, що передається пакетом $PCKT_q$. Об'єктом інтересу є дані, які можуть прямо або опосередковано вказувати на адресата, роль частки, порядок її використання або належність пакета до певного процесу передавання.

У запропонованому протоколі службова інформація розділена за призначенням: маршрутний блок X_q^L використовується проміжними вузлами для локального пересилання, а службовий блок X_q^R призначений кінцевому отримувачу. Тому проміжний вузол може отримати лише дані, необхідні для наступного переходу, але не повинен отримувати службовий контекст відновлення. Без розкриття X_q^R зловмисник не має підтверджених даних про кінцевого отримувача, порядок використання частки та її зв'язок з іншими частками. Отже, спроба встановити R за службовими даними обмежується структурою інкапсуляції.

Сценарій 8. Встановлення маршруту передавання.

У цьому сценарії розглядаються можливості зловмисників A1, A2, A4 та A5, які намагаються визначити маршрут L_q проходження пакету $PCKT_q$ через пірингову мережу. Залежно від класу зловмисника йому можуть бути доступні окремі мережеві події, локальні переходи між вузлами H_{mn} , частини маршрутного блоку X_q^L , час проходження пакету та адреси суміжних вузлів. Метою атаки є не обов'язково отримання вмісту частки, а побудова послідовності вузлів, через які

пакет рухається від відправника S до кінцевого отримувача R .

Основне обмеження для такого сценарію полягає в локальності маршрутної інформації. Окремий проміжний вузол не володіє повним маршрутом L_q , а отримує лише ті службові дані, які потрібні для виконання наступного переходу. Тому компрометація одного вузла або спостереження за однією ділянкою маршруту дає лише фрагмент шляху. Коаліція вузлів може зіставляти кілька таких фрагментів, однак без доступу до повного службового контексту та без охоплення достатньої кількості ділянок маршруту вона не отримує гарантованої повної послідовності передавання.

Отже, результат атаки на маршрут залежить від обсягу мережевої видимості зловмисника. Для $A1$ і $A4$ він обмежується локальним фрагментом маршруту. Для $A2$ і $A5$ можливе часткове трасування або побудова припущення щодо більшої частини L_q . Проте встановлення маршруту однієї частки не дорівнює встановленню всієї взаємодії $S \rightarrow R$, оскільки інші частки можуть передаватися іншими маршрутами, а службові дані отримувача залишаються недоступними для проміжних вузлів.

Сценарій 9. Sybil-атака на розподілене середовище передавання.

У цьому сценарії розглядається зловмисник $A6$, який створює або контролює значну кількість логічних вузлів у піринговій мережі. Його мета полягає у збільшенні ймовірності того, що контрольовані вузли потраплять до маршрутів передавання часток. На відміну від компрометації вже наявного проміжного вузла, Sybil-атака спрямована на розширення присутності зловмисника в самій множині вузлів, з якої формуються маршрути L_q .

У межах такого сценарію зловмисник може отримати більше локальних маршрутних спостережень, якщо його вузли частіше включаються до маршрутів проходження $PCKT_q$. Це підвищує можливість зіставлення маршрутних

фрагментів, часових ознак і належності окремих пакетів до одного процесу передавання. Водночас навіть контроль значної кількості вузлів не надає автоматичного доступу до службового блока отримувача X_q^R , параметрів відновлення, повної множини часток або параметрів розшифрування C .

Для обмеження такого сценарію суттєвими є правила формування маршрутів, контроль включення вузлів до пірингової мережі, обмеження повторного використання одних і тих самих вузлів та зменшення залежності маршрутизації від вузлів, контрольованих одним суб'єктом.

Сценарій 10. Активний вплив на пакети з метою деанонімізації.

У цьому сценарії розглядаються можливості зловмисників А3 та А8, зокрема їх активний вплив на передавання пакетів. До таких дій належать затримка, блокування, дублювання, повторне передавання або модифікація $PCKT_q$ чи його окремих складових. Метою атаки є не тільки порушення доступності, а й створення додаткових ознак, за якими зловмисник може надалі зіставити події в різних частинах маршруту, перевірити належність пакета до певного процесу передавання або пов'язати реакцію отримувача з конкретною дією над пакетом.

У разі затримки або вибіркового блокування пакета зловмисник може намагатися виявити відповідні зміни в подальшому проходженні трафіку. У разі дублювання або повторного передавання він може створювати додаткові мережеві події для перевірки того, чи буде пакет прийнятий у межах поточного процесу передавання. У разі модифікації частки P_q атака може використовуватися як спроба активного маркування пакета. Для такого сценарію важливими є обчислення геш-значень часток і перевірка службового контексту: зміна байтового вмісту частки має виявлятися за геш-значенням, а пакет, що не відповідає очікуваним службовим параметрам передавання, не повинен використовуватися під час відновлення.

Отже, активний вплив на пакети може створювати умови для деанонімізації через часові, маршрутні або поведінкові ознаки, однак не надає зловмиснику

прямого доступу до службового блоку отримувача, параметрів відновлення або відкритого вмісту зображення. У межах запропонованого протоколу така атака обмежується використанням геш-значень для контролю цілісності часток, інкапсуляцією службових даних і пороговою умовою відновлення. При цьому дії, пов'язані із затримкою або блокуванням пакетів, залишаються насамперед атаками на доступність передавання, а їхній вплив на анонімізацію проявляється через можливість створення ознак для подальшої кореляції.

Розглянуті сценарії показують, що основні атаки на анонімність передавання часток спрямовані не на безпосереднє розкриття вмісту зображення, а на встановлення зв'язків між мережевими подіями, маршрутами, частками, відправником S і кінцевим отримувачем R . Локальне спостереження та компрометація одного проміжного вузла дають лише обмежений фрагмент інформації про передавання. Компрометація кількох вузлів, часово-об'ємна кореляція, зв'язування часток і Sybil-атака підвищують можливість деанонімізації, оскільки дозволяють накопичувати та зіставляти маршрутні ознаки.

Запропонований протокол обмежує інформативність таких атак за рахунок розділення даних між частками, передавання їх через різні маршрути пірингової мережі, локальності маршрутного блока X_q^L , захисту службового блока отримувача X_q^R , використання геш-значень та порогової умови відновлення. Тому окремий пакет, окремий проміжний вузол або окремий маршрутний фрагмент не містять достатньої інформації для встановлення повної взаємодії $S \rightarrow R$ і не дають доступу до відкритого вмісту зображення.

4.6 Висновки до четвертого розділу

У четвертому розділі виконано експериментальне дослідження запропонованих методів шифрування та розподілу секретного вмісту зображень, а також проведено аналіз стійкості протоколу анонімної передачі часток через

пiрингову мережу. Отриманi результати пiдтверджують коректнiсть розроблених методiв та заявленi властивостi реалiзованих методiв.

Результати дослiдження статистичних властивостей зашифрованих зображень показали, що запропонований метод забезпечує суттєве перетворення структури початкових даних. Для зашифрованих зображень спостерiгається вирiвнювання розподiлу байтових значень, наближення ентропiї до граничного значення та зменшення кореляцiйних залежностей мiж сусiднiми елементами. Це свiдчить про зниження статистичної подiбності мiж початковим i зашифрованим поданням зображення та пiдтверджує ефективнiсть шифрування.

Оцiнки продуктивностi програмних засобiв показали, що порядок виконання операцiй впливає на часовi характеристики методу. Варiант PS продемонстрував вищу пропускну здатнiсть порiвняно з варiантом SP для всiх дослiджених обсягiв даних, що обгрунтовує його використання як основного варiанта реалiзацiї у подальших порiвняннях. Отриманi часовi характеристики мають переважно лiнiйну залежнiсть вiд обсягу вхiдного вектора, що узгоджується з алгоритмiчною природою запропонованого методу та пiдтверджує можливiсть його застосування для зображень рiзного розмiру.

Порiвняння з локальними реалiзацiями вiдомих методiв шифрування зображень показало, що запропонований метод має конкурентнi часовi характеристики. При цьому для реалiзацiї методу не потрiбно зберiгати таблицi перестановки та використовувати складнi багатораундовi або алгебраїчнi перетворення, що є важливим для практичної реалiзацiї в системах, де обробляються великi обсяги графiчних даних.

Експериментальнi дослiдження запропонованих схем розподiлу пiдтвердили коректнiсть формування та вiдновлення часток. У зазначених конфiгурацiях вiдновлений зашифрований вектор точно вiдповiдає початковому зашифрованому

вектору, що забезпечує подальше розшифрування та відновлення зображення без помилок.

Параметри (k, n) схеми розподілу безпосередньо впливають на сумарний обсяг розподілених даних і практичну вартість зберігання або передавання. Це дозволяє розглядати вибір параметрів схеми як окремий елемент налаштування системи: збільшення надлишковості може підвищувати стійкість до втрати частини даних, тоді як зменшення сумарного обсягу часток є важливим для ефективного використання ресурсів.

Аналіз стійкості протоколу анонімної передачі зображення показав, що основною характеристикою для цього сценарію є анонімізація взаємодії між відправником і кінцевим отримувачем. Розглянуті сценарії атак засвідчують, що локальне спостереження, перехоплення окремого пакета або компрометація одного проміжного вузла не забезпечують зловмиснику повної інформації про маршрут, службовий контекст відновлення або зв'язок між S і R . Більш суттєвими для анонімності є сценарії, пов'язані з розширеною видимістю мережевих подій, компрометацією кількох вузлів, часово-об'ємною кореляцією та Sybil-атака на маршрутизацію.

Встановлено, що стійкість протоколу формується не одним окремим механізмом, а взаємодією кількох складових: попереднього зашифрування зображення, розподілу зашифрованого вектора на частки, використання дилерської частки D , геш-значень часток, ключової інкапсуляції службових даних і багатомаршрутного передавання через пірингову мережу. Завдяки цьому складові мають обмежену інформативність і не надають достатніх даних для відновлення відкритого зображення або повної деанонімізації передавання.

Висновки

У дисертаційній роботі розв'язано науково-прикладне завдання пришвидшення процесів криптографічного захисту секретного вмісту зображень шляхом розроблення методів шифрування і розподілу секрету та засобів, що їх реалізують. У межах роботи одержано теоретичні та експериментальні результати, що підтверджують зменшення кількості базових операцій під час шифрування, скорочення часу формування і відновлення часток у запропонованих схемах розподілу, зменшення сумарного обсягу індивідуальних часток у (k,n) -схемах та формування зашифрованих даних із належними статистичними властивостями.

Проведено аналіз сучасних підходів до криптографічного захисту зображень, зокрема методів шифрування, схем розподілу секрету та комбінованих підходів. Встановлено, що відомі методи шифрування зображень часто передбачають багатораундову обробку, складні арифметичні операції, використання хаотичних систем або додаткові перетворення подання зображення. Схеми розподілу секрету забезпечують формалізовані умови доступу, однак у задачах захисту зображень можуть потребувати значної кількості обчислень або формувати частки з надлишковим сумарним обсягом. Це обґрунтовує доцільність розроблення методів, для яких визначальними є кількість базових операцій, час виконання процедур шифрування, розподілу та відновлення, обсяг сформованих часток і статистичні властивості зашифрованих даних.

Розроблено метод шифрування зображень, який використовує побайтову підстановку та перестановку байтів у межах усього зображення. Початкове зображення подається у вигляді байтового вектора, що дозволяє виконувати обробку незалежно від формату графічних даних. Для формування перестановки використано РЗЛЗЗ, поділ простору індексів на піддіапазони та систему лічильників. Така організація забезпечує алгоритмічне формування порядку розміщення байтів без зберігання повної таблиці перестановки.

Для методу шифрування розглянуто два варіанти реалізації — SP та PS. У варіанті SP підстановка і перестановка виконуються послідовно, тоді як у варіанті PS ці операції поєднуються в одному проході. Теоретична оцінка показала, що PS-варіант потребує 18 базових операцій на один байт, тоді як для розглянутих відомих методів цей показник становить від 101 до 1074 операцій на байт. Отже, запропонований PS-варіант потребує у 5,6–59,6 рази менше базових операцій порівняно з дослідженими методами шифрування зображень. Експериментальне порівняння SP- та PS-варіантів також показало перевагу PS-варіанту за пропускну здатністю приблизно у 1,81 рази.

Удосконалено метод розподілу секретного вмісту зображень за (n,n) -схемою шляхом використання попередньо зашифрованого зображення та циклічного формування часток зашифрованого вектора. Розподіл виконується через послідовне призначення байтів за індексом, що спрощує процедури формування часток і відновлення. Така схема забезпечує поєднання попереднього шифрування з операційно простою процедурою розподілу, де окремі частки містять лише елементи зашифрованого вектора.

Розроблено коаліційну (k,n) -схему розподілу секретного вмісту зображень, у якій частки формуються з блоків попередньо зашифрованого зображення відповідно до допустимих коаліцій учасників, а відсутня складова відновлюється з використанням частки дилера. Також розроблено спрощену (k,n) -схему з перекриттям блоків, у якій формування часток виконується за циклічним правилом без побудови множини коаліцій. Спрощена схема зменшує складність організації розподілу порівняно з коаліційною схемою за рахунок регулярної структури формування часток.

Виконано оцінку обсягу часток у запропонованих (k,n) -схемах. Встановлено, що коаліційна та спрощена схеми забезпечують зменшення сумарного обсягу індивідуальних часток у 1,25–5 разів порівняно з розглянутими схемами розподілу

секрету. Це підтверджує перевагу запропонованих схем за обсягом даних, які зберігаються учасниками, без необхідності формування повнорозмірних або надмірно збільшених індивідуальних часток.

Розроблено програмні засоби, що реалізують запропоновані методи шифрування, розподілу та відновлення зображень. Програмна реалізація охоплює SP- та PS-варіанти шифрування, (n,n) -схему розподілу, коаліційну (k,n) -схему та спрощену (k,n) -схему з перекриттям блоків. Це дало змогу виконати експериментальне дослідження запропонованих методів, порівняти їх із відомими підходами та перевірити узгодженість практичних результатів із теоретичними оцінками.

Запропоновано практичні сценарії застосування розроблених (k,n) -схем. Коаліційну схему використано як основу протоколу анонімної передачі секретного вмісту зображення у пірінговій мережі, де частки передаються різними маршрутами через проміжні вузли. Спрощену схему з перекриттям блоків використано для протоколу захищеного зберігання чутливих медичних даних. Аналіз протоколу анонімної передачі показав, що розподіл часток між різними маршрутами, використання службових зашифрованих даних і відсутність повного вмісту зображення в окремого проміжного вузла ускладнюють встановлення зв'язку між відправником, отримувачем і переданими частками.

Проведено експериментальні дослідження статистичних властивостей зашифрованих зображень. Значення ентропії зашифрованих даних перебувають у діапазоні 7,9993–7,9999, що наближається до граничного значення для 8-бітного подання та вказує на рівномірність розподілу значень після шифрування. Кореляційні залежності між сусідніми елементами після шифрування зменшено до значень порядку 10^{-3} - 10^{-5} , що свідчить про руйнування статистичних залежностей, характерних для початкових зображень. Результати тестування за NIST SP 800-22 показали, що сформовані бітові послідовності не мають виражених

закономірностей для досліджених конфігурацій методу. Таким чином, експериментальні результати підтверджують, що запропоноване шифрування формує дані з властивостями, близькими до випадкового розподілу, що є необхідним для подальшого безпечного розподілу зашифрованого вмісту між частками.

Проведено експериментальне оцінювання часових характеристик запропонованих схем розподілу. Порівняно з дослідженими схемами розподілу секрету запропоновані схеми забезпечують скорочення часу формування часток приблизно у 9–21 раз і часу відновлення приблизно у 33–115 разів. Отримані результати підтверджують перевагу запропонованих схем за часовими характеристиками процедур розподілу та відновлення секретного вмісту зображень.

Отже, у дисертаційній роботі виконано всі поставлені завдання: проаналізовано сучасні підходи до шифрування та розподіленого зберігання зображень; розроблено метод шифрування зображень; розроблено методи розподілу секретного вмісту зображень за (n,n) - і (k,n) -схемами; розроблено програмні засоби, що реалізують запропоновані методи; проведено експериментальні дослідження їх властивостей. Сукупність отриманих результатів підтверджує досягнення мети роботи — пришвидшення процесів шифрування та розподілу секретного вмісту зображень за рахунок зменшення кількості базових операцій, скорочення часу виконання процедур розподілу та відновлення, а також зменшення сумарного обсягу індивідуальних часток у запропонованих (k,n) -схемах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Beimel A., Farràs O., Lasri O. Improved Polynomial Secret-Sharing Schemes. *Theory of Cryptography. TCC 2023. Lecture Notes in Computer Science*. Vol. 14370. Cham : Springer, 2023. P. 374–405. DOI: 10.1007/978-3-031-48618-0_13.
2. Patel S., Persiano G., Seo J. Y., Yeo K. Efficient Secret Sharing for Large-Scale Applications. *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*. New York : ACM, 2024. P. 3065–3079. DOI: 10.1145/3658644.3670379.
3. Cascudo I., Cozzo D., Giunta E. Verifiable Secret Sharing from Symmetric Key Cryptography with Improved Optimistic Complexity. *Advances in Cryptology — ASIACRYPT 2024. Lecture Notes in Computer Science*. Cham : Springer, 2024. P. 100–128. DOI: 10.1007/978-981-96-0941-3_4.
4. Dziembowski S., Faust S., Lizeur T., Mielniczuk M. Secret Sharing with Snitching. *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*. New York : ACM, 2024. P. 840–853. DOI: 10.1145/3658644.3690296.
5. Holla M. R., Suma D., Pais A. R. Accelerating randomized image secret sharing with GPU: contrast enhancement and secure reconstruction using progressive and convolutional approaches. *Multimedia Tools and Applications*. 2024. Vol. 83. P. 43761–43776. DOI: 10.1007/s11042-024-18634-w.
- [6] Zia S. M. U., McCartney M., Scotney B., Martinez Carracedo J., AbuTair M., Memon J., Sajjad A. Survey on Image Encryption Techniques using Chaotic Maps in Spatial, Transform and Spatiotemporal Domains. *International Journal of Information Security*. 2022. Vol. 21. P. 917–935. DOI: 10.1007/s10207-022-00588-5.
- [7] Hazzazi M. M., Rehman M. U., Shafique A., Aljaedi A., Bassfar Z., Usman A. B. Enhancing image security via chaotic maps, Fibonacci, Tribonacci transformations, and DWT diffusion: a robust data encryption approach. *Scientific Reports*. 2024. Vol. 14. Article 12277. DOI: 10.1038/s41598-024-62260-3.
- [8] Umar T. Chaos based image encryption scheme to secure sensitive data. *Expert Systems with Applications*. 2024. Article 125050. DOI: 10.1016/j.eswa.2024.125050.
- [9] Nechyporuk M., Luzhetskyyi V., Romanyuk O. Accelerated encryption of graphic images. *Agriculture, technical and applied sciences: main problems and solutions : collective monograph*. Boston : Primedia eLaunch, 2025. Chap. 8.1. P. 218–232. DOI: 10.46299/ISG.2025.MONO.TECH.1.
- [10] Avramenko V., Bondarenko M. Image cryptosystem with image key using integral disproportion. *Radioelectronic and Computer Systems*. 2024. No. 2(110). P. 147–159. DOI: 10.32620/reks.2024.2.12.

- [11] Dmitrishin D. V., Khamitov V. M., Antoshchuk S. G., Boltenkov V. O. A modified image encryption algorithm based on the chaotic Tent map. *Herald of Advanced Information Technology*. 2026. Vol. 9, No. 1. P. 9–19. DOI: [10.15276/hait.09.2026.01](https://doi.org/10.15276/hait.09.2026.01).
- [12] Khamitov V. M., Boltenkov V. O., Antoshchuk S. G. A method for operative sharing confidential images among a group of trusted users. *Applied Aspects of Information Technology*. 2026. Vol. 9, No. 2. P. 200–207. DOI: [10.15276/aait.09.2026.14](https://doi.org/10.15276/aait.09.2026.14).
- [13] Онацький О. В., Дикий О. В., Жарова О. В., Йона Л. Г. Модифікація порогових схем розподілу секрету Шаміра та Блеклі на еліптичних кривих. *Інфокомунікаційні та комп'ютерні технології*. 2022. Т. 1, № 03. С. 267–283. DOI: [10.36994/2788-5518-2022-01-03-16](https://doi.org/10.36994/2788-5518-2022-01-03-16).
- [14] Palagin O. O., Sokolov A. V. The algorithm for encryption of graphic information based on chaotic map and Galois field transform. *Informatics and Mathematical Methods in Simulation*. 2023. Vol. 13, No. 3–4. P. 204–211. DOI: [10.15276/imms.v13.no3-4.204](https://doi.org/10.15276/imms.v13.no3-4.204).
- [15] Майданюк В. П., Романюк О. Н., Павлов С. В., Нечипорук М. Л. Підвищення коефіцієнта ущільнення зображень на основі двовимірних ортогональних перетворень. *Наукові праці ДонНТУ. Серія “Інформатика, кібернетика та обчислювальна техніка”*. 2024. № 2(39). С. 31–40. DOI: [10.31474/1996-1588-2024-2-39-31-40](https://doi.org/10.31474/1996-1588-2024-2-39-31-40)
- [16] Shamir A. How to share a secret. *Communications of the ACM*. 1979. Vol. 22, No. 11. P. 612–613. DOI: [10.1145/359168.359176](https://doi.org/10.1145/359168.359176).
- [17] Wu X., Yang C.-N., Yang Y.-Y. A hybrid scheme for enhancing recovered image quality in polynomial based secret image sharing by modify-and-recalculate strategy. *Journal of Information Security and Applications*. 2020. Vol. 51. Article 102452. DOI: [10.1016/j.jisa.2020.102452](https://doi.org/10.1016/j.jisa.2020.102452).
- [18] Luo S., Liu Y., Yan X., Yu Y. Secret image sharing scheme with lossless recovery and high efficiency. *Signal Processing*. 2023. Vol. 206. Article 108931. DOI: [10.1016/j.sigpro.2023.108931](https://doi.org/10.1016/j.sigpro.2023.108931).
- [19] Xie D., Hu Z., Zhang Y., Wang T., Chen F. Reversible essential image secret sharing. *Optics & Laser Technology*. 2024. Vol. 170. Article 110232. DOI: [10.1016/j.optlastec.2023.110232](https://doi.org/10.1016/j.optlastec.2023.110232).
- [20] Sun Q., Cao H., Li S., Song H., Liu Y. Changing the Threshold in a Bivariate Polynomial Based Secret Image Sharing Scheme. *Mathematics*. 2022. Vol. 10, No. 5. Article 710. DOI: [10.3390/math10050710](https://doi.org/10.3390/math10050710).
- [21] Maity K., Mukhopadhyay S. LPSIS: a lossless secret image sharing scheme based on Legendre polynomials with low-cost reconstruction. *The Visual Computer*. 2025. Vol. 41, No. 7. P. 4623–4637. DOI: [10.1007/s00371-024-03681-z](https://doi.org/10.1007/s00371-024-03681-z).
- [22] Li X., Fu Z., Gao Y., Yu B. An enhanced image protection framework: polynomial-based secret sharing meets information hiding. *Cybersecurity*. 2025. Vol. 8. Article 129. DOI: [10.1007/s44443-025-00143-x](https://doi.org/10.1007/s44443-025-00143-x).

- [23] Ding J., Ke P., Lin C., Wang H. Bivariate polynomial-based secret sharing schemes with secure secret reconstruction. *Information Sciences*. 2022. Vol. 593. P. 398–414. DOI: [10.1016/j.ins.2022.02.005](https://doi.org/10.1016/j.ins.2022.02.005).
- [24] Blakley G. R. Safeguarding cryptographic keys. *Proceedings of the AFIPS National Computer Conference*. 1979. Vol. 48. P. 313–317. DOI: [10.1145/800135.804486](https://doi.org/10.1145/800135.804486).
- [25] Beimel A. Secret-sharing schemes: A survey. *Coding and Cryptology: IWCC 2011. Lecture Notes in Computer Science*. 2011. Vol. 6639. P. 11–46. DOI: [10.1007/978-3-642-22351-8_1](https://doi.org/10.1007/978-3-642-22351-8_1).
- [26] Ghebleh M., Kanso A., Alazemi A. A Linear Algebraic Threshold Essential Secret Image Sharing Scheme. *Mathematics*. 2022. Vol. 10, No. 5. Article 742. DOI: [10.3390/math10050742](https://doi.org/10.3390/math10050742).
- [27] Bhattacharjee T., Rout R. K., Maity S. P. Affine Boolean classification in secret image sharing for progressive quality access control. *Journal of Information Security and Applications*. 2017. Vol. 33. P. 16–29. DOI: [10.1016/j.jisa.2017.01.003](https://doi.org/10.1016/j.jisa.2017.01.003).
- [28] Sun Y., Lu Y., Yan X., Liu L., Li L. Robust Secret Image Sharing Scheme Against Noise in Shadow Images. *IEEE Access*. 2021. Vol. 9. P. 23284–23300. DOI: [10.1109/ACCESS.2021.3056893](https://doi.org/10.1109/ACCESS.2021.3056893).
- [29] Ding C., Yuan J. Covering and Secret Sharing with Linear Codes. *Discrete Mathematics and Theoretical Computer Science. Lecture Notes in Computer Science*. 2003. Vol. 2731. P. 11–25. DOI: [10.1007/3-540-45066-1_2](https://doi.org/10.1007/3-540-45066-1_2).
- [30] Mignotte M. How to share a secret. *Advances in Cryptology — EUROCRYPT 1982. Lecture Notes in Computer Science*. Berlin: Springer, 1983. Vol. 149. P. 371–375. DOI: [10.1007/3-540-39799-X_1](https://doi.org/10.1007/3-540-39799-X_1).
- [31] Asmuth C., Bloom J. A modular approach to key safeguarding. *IEEE Transactions on Information Theory*. 1983. Vol. 29, No. 2. P. 208–210. DOI: [10.1109/TIT.1983.1056650](https://doi.org/10.1109/TIT.1983.1056650).
- [32] Xiong L., Han X., Yang C.-N. CP-PSIS: CRT and polynomial-based progressive secret image sharing. *Signal Processing*. 2021. Vol. 185. Article 108064. DOI: [10.1016/j.sigpro.2021.108064](https://doi.org/10.1016/j.sigpro.2021.108064).
- [33] Ke Y., Zhang M., Zhang X., Liu J., Su T., Yang X. A reversible data hiding scheme in encrypted domain for secret image sharing based on Chinese remainder theorem. *IEEE Transactions on Circuits and Systems for Video Technology*. 2022. Vol. 32, No. 4. P. 2469–2481. DOI: [10.1109/TCSVT.2021.3081575](https://doi.org/10.1109/TCSVT.2021.3081575).
- [34] Sharobim B. K., Fetteha M. A., Abd-El-Hafiz S. K., Sayed W. S., Said L. A., Radwan A. G. An Efficient Multi-Secret Image Sharing System Based on Chinese Remainder Theorem and Its FPGA Realization. *IEEE Access*. 2023. DOI: [10.1109/ACCESS.2023.3240202](https://doi.org/10.1109/ACCESS.2023.3240202).
- [35] Li L., Liu J., Liu X., Zhang X. Practical Secret Image Sharing Based on the Chinese Remainder Theorem. *Mathematics*. 2022. Vol. 10, No. 12. Article 1959. DOI: [10.3390/math10121959](https://doi.org/10.3390/math10121959).

- [36] Deshmukh M., Nain N., Ahmed M. A novel approach for sharing multiple color images by employing Chinese remainder theorem. *Journal of Visual Communication and Image Representation*. 2017. Vol. 49. P. 291–302. DOI: [10.1016/j.jvcir.2017.09.013](https://doi.org/10.1016/j.jvcir.2017.09.013).
- [37] Gonzalez, R.C. and Woods, R.E. (2018) *Digital Image Processing*. 4th Edition, Pearson Education, New York, 1022 p.
- [38] Zia S. M. U., McCartney M., Scotney B., Martinez Carracedo J., AbuTair M., Memon J., Sajjad A. Survey on Image Encryption Techniques using Chaotic Maps in Spatial, Transform and Spatiotemporal Domains. *International Journal of Information Security*. 2022. Vol. 21, No. 4. P. 917–935. DOI: [10.1007/s10207-022-00588-5](https://doi.org/10.1007/s10207-022-00588-5).
- [39] Daemen J., Rijmen V. *The Design of Rijndael: AES — The Advanced Encryption Standard*. Berlin: Springer, 2002. DOI: [10.1007/978-3-662-04722-4](https://doi.org/10.1007/978-3-662-04722-4).
- [40] Zhang Y. Test and Verification of AES Used for Image Encryption. *3D Research*. 2018. Vol. 9, No. 1. DOI: [10.1007/s13319-017-0154-7](https://doi.org/10.1007/s13319-017-0154-7).
- [41] Arab A., Rostami M. J., Ghavami B. An image encryption method based on chaos system and AES algorithm. *The Journal of Supercomputing*. 2019. Vol. 75. P. 6663–6682. DOI: [10.1007/s11227-019-02878-7](https://doi.org/10.1007/s11227-019-02878-7).
- [42] Taha M. D. Parallel AES Image Encryption with Logistic Map. *Passer Journal of Basic and Applied Sciences*. 2025. Vol. 7, No. 2. P. 707–716. DOI: [10.24271/psr.2025.509863.1986](https://doi.org/10.24271/psr.2025.509863.1986).
- [43] Zhang S. An Image Encryption Method based on Pixel-level Diffusion and Bit Flip. *ACM International Conference Proceedings*. 2021. DOI: [10.1145/3494885.3494889](https://doi.org/10.1145/3494885.3494889).
- [44] Pareek N. K., Patidar V., Sud K. K. Image encryption using chaotic logistic map. *Image and Vision Computing*. 2006. Vol. 24, No. 9. P. 926–934. DOI: [10.1016/j.imavis.2005.08.007](https://doi.org/10.1016/j.imavis.2005.08.007).
- [45] Lian S., Sun J., Wang Z. A block cipher based on chaotic maps. *Chaos, Solitons & Fractals*. 2005. Vol. 26, No. 1. P. 117–129. DOI: [10.1016/j.chaos.2004.09.012](https://doi.org/10.1016/j.chaos.2004.09.012).
- [46] Zhu S. A new robust chaotic map for image encryption. *Mathematics and Computers in Simulation*. 2023. DOI: [10.1016/j.matcom.2023.01.012](https://doi.org/10.1016/j.matcom.2023.01.012).
- [47] Zhang B. Chaos-Based Image Encryption: Review, Application, and Challenges. *Mathematics*. 2023. Vol. 11, No. 11. Article 2585. DOI: [10.3390/math11112585](https://doi.org/10.3390/math11112585).
- [48] Lorenz E. N. Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*. 1963. Vol. 20, No. 2. P. 130–141. DOI: [10.1175/1520-0469\(1963\)020<0130:DNF>2.0.CO;2](https://doi.org/10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2).
- [49] Chen G., Ueta T. Yet another chaotic attractor. *International Journal of Bifurcation and Chaos*. 1999. Vol. 9, No. 7. P. 1465–1466. DOI: [10.1142/S0218127499001024](https://doi.org/10.1142/S0218127499001024).
- [50] Lü J., Chen G. A new chaotic attractor coined. *International Journal of Bifurcation and Chaos*. 2002. Vol. 12, No. 3. P. 659–661. DOI: [10.1142/S0218127402004620](https://doi.org/10.1142/S0218127402004620).

- [51] Niu Y., Zhou H., Zhang X. Image encryption scheme based on improved four-dimensional chaotic system and evolutionary operators. *Scientific Reports*. 2024. Vol. 14. Article 7033. DOI: 10.1038/s41598-024-57756-x.
- [52] Zhang H., Liu X., Chen K., Te R., Yan F. Robust Image Encryption with 2D Hyperchaotic Map and Dynamic DNA-Zigzag Encoding. *Entropy*. 2025. Vol. 27, No. 6. Article 606. DOI: 10.3390/e27060606.
- [53] Hua Z., Jin F., Xu B., Huang H. 2D Logistic-Sine-Coupling Map for Image Encryption. *Signal Processing*. 2018. Vol. 149. P. 148–161. DOI: 10.1016/j.sigpro.2018.03.010.
- [54] Hua Z., Zhou Y., Huang H. Cosine-transform-based chaotic system for image encryption. *Information Sciences*. 2019. Vol. 480. P. 403–419. DOI: 10.1016/j.ins.2018.12.048.
- [55] Wen H., Lin Y., Kang S., Zhang X., Zou K. Secure image encryption algorithm using chaos-based block permutation and weighted bit planes chain diffusion. *iScience*. 2024. Vol. 27, No. 1. Article 108610. DOI: 10.1016/j.isci.2023.108610.
- [56] Huang X., Ye G. An efficient symmetric image encryption algorithm based on an intertwining logistic map. *Neurocomputing*. 2017. Vol. 251. P. 45–53. DOI: 10.1016/j.neucom.2017.04.016.
- [57] Zhang B. Chaos-Based Image Encryption: Review, Application, and Challenges. *Mathematics*. 2023. Vol. 11, No. 11. Article 2585. DOI: 10.3390/math11112585.
- [58] Belazi A., Abd El-Latif A. A., Belghith S. A novel image encryption scheme based on substitution-permutation network and chaos. *Signal Processing*. 2016. Vol. 128. P. 155–170. DOI: [10.1016/j.sigpro.2016.03.021](https://doi.org/10.1016/j.sigpro.2016.03.021).
- [59] Chai X., Gan Z., Yuan K., Chen Y., Liu X. A novel image encryption scheme based on DNA sequence operations and chaotic systems. *Neural Computing and Applications*. 2019. DOI: [10.1007/s00521-017-2993-9](https://doi.org/10.1007/s00521-017-2993-9).
- [60] Li C., Lin D., Feng B., Lü J., Hao F. Cryptanalysis of a chaotic image encryption algorithm based on information entropy. *IEEE Access*. 2018. Vol. 6. P. 75834–75842. Article 8548554. DOI: 10.1109/ACCESS.2018.2883690.
- [61] Preishuber M., Hütter T., Katzenbeisser S., Uhl A. Depreciating Motivation and Empirical Security Analysis of Chaos-Based Image and Video Encryption. *IEEE Transactions on Information Forensics and Security*. 2018. Vol. 13, No. 9. P. 2137–2150. DOI: 10.1109/TIFS.2018.2812080.
- [62] Lin Y., Xie Z., Chen T., Cheng X., Wen H. Image privacy protection scheme based on high-quality reconstruction DCT compression and nonlinear dynamics. *Expert Systems with Applications*. 2024. Vol. 257. Article 124891. DOI: [10.1016/j.eswa.2024.124891](https://doi.org/10.1016/j.eswa.2024.124891).
- [63] Réfrégier P., Javidi B. Optical image encryption based on input plane and Fourier plane random encoding. *Optics Letters*. 1995. Vol. 20, No. 7. P. 767–769. DOI: [10.1364/OL.20.000767](https://doi.org/10.1364/OL.20.000767).

- [64] Khan J. S., Ahmad J., Khan M. A. Chaos based efficient selective image encryption. *Multidimensional Systems and Signal Processing*. 2019. Vol. 30. P. 943–961. DOI: [10.1007/s11045-018-0589-x](https://doi.org/10.1007/s11045-018-0589-x).
- [65] An F.-P., Liu J.-E. Image Encryption Algorithm Based on Adaptive Wavelet Chaos. *Journal of Sensors*. 2019. Article 2768121. DOI: [10.1155/2019/2768121](https://doi.org/10.1155/2019/2768121).
- [66] Zhong H., Wang G., Wu X., Jiao J. Multi-image encryption algorithm based on wavelet transform. *Multimedia Tools and Applications*. 2022. DOI: [10.1007/s11042-022-12479-x](https://doi.org/10.1007/s11042-022-12479-x).
- [67] Huang X., Dong Y., Ye G., Shi Y. Meaningful image encryption algorithm based on compressive sensing and integer wavelet transform. *Frontiers of Computer Science*. 2023. Vol. 17. Article 173804. DOI: [10.1007/s11704-022-1419-8](https://doi.org/10.1007/s11704-022-1419-8).
- [68] Li Y.-M., Jiang M., Wei D., Deng Y. A Novel Image Encryption Algorithm Based on Compressive Sensing and a Two-Dimensional Linear Canonical Transform. *Fractal and Fractional*. 2024. Vol. 8, No. 2. Article 92. DOI: [10.3390/fractalfract8020092](https://doi.org/10.3390/fractalfract8020092).
- [69] Li Y.-M., Jiang M., Wei D., Deng Y. Fast Encryption Algorithm Based on Chaotic System and Cyclic Shift in Integer Wavelet Domain. *Fractal and Fractional*. 2024. Vol. 8, No. 2. Article 75. DOI: [10.3390/fractalfract8020075](https://doi.org/10.3390/fractalfract8020075).
- [70] George L. E., Hassan E. Kh., Mohammed S. G., Mohammed F. G. Selective Image Encryption Based on DCT, Hybrid Shift Coding and Randomly Generated Secret Key. *Iraqi Journal of Science*. 2020. Vol. 61, No. 4. P. 920–935. DOI: [10.24996/ijis.2020.61.4.25](https://doi.org/10.24996/ijis.2020.61.4.25).
- [71] Hua Z., Zhou Y., Huang H. Cosine-transform-based chaotic system for image encryption. *Information Sciences*. 2019. Vol. 480. P. 403–419. DOI: [10.1016/j.ins.2018.12.048](https://doi.org/10.1016/j.ins.2018.12.048).
- [72] Wen H., Lin Y., Kang S., Zhang X., Zou K. Secure image encryption algorithm using chaos-based block permutation and weighted bit planes chain diffusion. *iScience*. 2024. Vol. 27. Article 108610. DOI: [10.1016/j.isci.2023.108610](https://doi.org/10.1016/j.isci.2023.108610).
- [73] Wang R., Yang G., Yan X., Luo S., Han Q. Secret image sharing in the encrypted domain. *Journal of Visual Communication and Image Representation*. 2024. Vol. 98. Article 104013. DOI: [10.1016/j.jvcir.2023.104013](https://doi.org/10.1016/j.jvcir.2023.104013).
- [74] Lin J.-Y., Horng J.-H., Chang C.-C. Secret image sharing with distinct covers based on improved Cycling-XOR. *Journal of Visual Communication and Image Representation*. 2024. Vol. 104. Article 104282. DOI: [10.1016/j.jvcir.2024.104282](https://doi.org/10.1016/j.jvcir.2024.104282).
- [75] Gao H., Zheng G., Gao T. A secure image encryption-sharing based on different diffusion intensities by hyper-chaos according to image features. *Signal, Image and Video Processing*. 2024. Vol. 18, No. 11. P. 8119–8130. DOI: [10.1007/s11760-024-03455-z](https://doi.org/10.1007/s11760-024-03455-z).
- [76] Liang J., Peng H., Li L. A secure and effective image encryption scheme by combining parallel compressed sensing with secret sharing scheme. *Journal of*

- Information Security and Applications. 2023. Vol. 75. Article 103487. DOI: [10.1016/j.jisa.2023.103487](https://doi.org/10.1016/j.jisa.2023.103487).
- [77] Xie Y., Yu J., Guo S., Ding Q., Wang E. Image Encryption Scheme with Compressed Sensing Based on New Three-Dimensional Chaotic System. *Entropy*. 2019. Vol. 21, No. 9. Article 819. DOI: [10.3390/e21090819](https://doi.org/10.3390/e21090819).
- [78] Yu J., Guo S., Song X., Xie Y., Wang E. Parallel Mixed Image Encryption and Extraction Algorithm Based on Compressed Sensing. *Entropy*. 2021. Vol. 23, No. 3. Article 278. DOI: [10.3390/e23030278](https://doi.org/10.3390/e23030278).
- [79] Wu B., Xie D., Chen F., Zhu H., Wang X., Zeng Y. Compressed sensing based visually secure multi-secret image encryption-sharing scheme. *Multimedia Tools and Applications*. 2023. DOI: [10.1007/s11042-023-15922-9](https://doi.org/10.1007/s11042-023-15922-9).
- [80] Xie D., Wu B., Chen F., Wang T., Hu Z., Zhang Y. A low-overhead compressed sensing-driven multi-party secret image sharing scheme. *Multimedia Systems*. 2023. Vol. 29. P. 1187–1202. DOI: [10.1007/s00530-023-01049-2](https://doi.org/10.1007/s00530-023-01049-2).
- [81] Wen H., Yang L., Bai C., Lin Y., Liu T., Chen L., Hu Y., He D. Exploiting high-quality reconstruction image encryption strategy by optimized orthogonal compressive sensing. *Scientific Reports*. 2024. Vol. 14. Article 8805. DOI: [10.1038/s41598-024-59277-z](https://doi.org/10.1038/s41598-024-59277-z).
- [82] Gonzalez R. C., Woods R. E. *Digital Image Processing*. 4th ed. Pearson, 2018.
- [83] ITU-T. Recommendation T.81: Information technology — Digital compression and coding of continuous-tone still images — Requirements and guidelines. 1992.
- [84] Portable Network Graphics (PNG) Specification (Third Edition). W3C Recommendation. World Wide Web Consortium, 2025. 24 June 2025. URL: <https://www.w3.org/TR/png-3/>.
- [85] TIFF Revision 6.0. Final Specification. Adobe Developers Association, 1992. 3 June 1992. URL: <https://www.itu.int/itudoc/itu-t/com16/tiff-fx/docs/tiff6.pdf>.
- [86] Shannon C. E. A Mathematical Theory of Communication. *The Bell System Technical Journal*. 1948. Vol. 27, No. 3. P. 379–423. DOI: <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>.
- [87] Лужецький В. А., Горбенко І. С. Методи шифрування на основі перестановки блоків змінної довжини. *Захист інформації*. 2015. Т. 17, № 2. С. 169–175.
- [88] Benaloh J., Leichter J. Generalized Secret Sharing and Monotone Functions. *Advances in Cryptology – CRYPTO’88. Lecture Notes in Computer Science*. Berlin, Heidelberg : Springer, 1990. Vol. 403. P. 27–35. DOI: [10.1007/0-387-34799-2_3](https://doi.org/10.1007/0-387-34799-2_3).
- [89] Ito M., Saito A., Nishizeki T. Secret sharing scheme realizing general access structure. *Electronics and Communications in Japan*. 1989. Vol. 72, No. 9. P. 56–64. DOI: [10.1002/ecjc.4430720906](https://doi.org/10.1002/ecjc.4430720906).
- [90] Кавка О. О., Майданюк В. П., Романюк О. Н., Завальнюк Є. К. Аналіз алгоритмів стиснення зображень із втратами. *Інформаційні технології та*

- комп'ютерна інженерія. 2023. Вип. 58, №3. С. 59-64. DOI: 10.31649/1999-9941-2023-58-3-59-64.
- [91] Майданюк В. П., Романюк О. Н., Павлов С. В., Нечипорук М. Л. Підвищення коефіцієнта ущільненні зображень на основі двовимірних ортогональних перетворень. Наукові праці ДонНТУ. Серія: Інформатика, кібернетика та обчислювальна техніка. 2024. №2(39). С. 31- 40.
- [92] National Institute of Standards and Technology. Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197-upd1. Gaithersburg, MD: NIST, 2001; updated 2023. DOI: <https://doi.org/10.6028/NIST.FIPS.197-upd1>.
- [93] Pareek N. K., Patidar V., Sud K. K. Image encryption using chaotic logistic map. Image and Vision Computing. 2006. Vol. 24, No. 9. P. 926–934. DOI: 10.1016/j.imavis.2005.08.007.
- [94] Soleymani A., Nordin M. J., Sundararajan E. A Chaotic Cryptosystem for Images Based on Henon and Arnold Cat Map. The Scientific World Journal. 2014. Article ID 536930. DOI: <https://doi.org/10.1155/2014/536930>.
- [95] Broumandnia A. Image encryption algorithm based on the finite fields in chaotic maps. Journal of Information Security and Applications. 2020. Vol. 54. Article 102553. DOI: <https://doi.org/10.1016/j.jisa.2020.102553>.
- [96] Patidar V., Kaur G. A novel conservative chaos driven dynamic DNA coding for image encryption. Frontiers in Applied Mathematics and Statistics. 2023. Vol. 8. Article 1100839. DOI: <https://doi.org/10.3389/fams.2022.1100839>.
- [97] Wen H., Lin Y., Kang S., Zhang X., Zou K. Secure image encryption algorithm using chaos-based block permutation and weighted bit planes chain diffusion. iScience. 2024. Vol. 27. Article 108610. DOI: 10.1016/j.isci.2023.108610.
- [98] Zhong H., Wang G., Wu X., Jiao J. Image Encryption Algorithm Based on a Novel Wide Range Hyperchaotic System and Image-Sensitive Function. Mathematics. 2022. Vol. 10, No. 15. Article 2583. DOI: <https://doi.org/10.3390/math10152583>.
- [99] NIST. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22 Revision 1a. 2010. DOI: [10.6028/NIST.SP.800-22r1a](https://doi.org/10.6028/NIST.SP.800-22r1a).
- [100] ISO/IEC 14882:2024. Programming languages — C++. International Organization for Standardization, 2024. URL: <https://www.iso.org/standard/83626.html>
- [101] Blakley G. R., Meadows C. Security of Ramp Schemes. Advances in Cryptology — CRYPTO 1984. Lecture Notes in Computer Science. Berlin: Springer, 1985. Vol. 196. P. 242–269. DOI: https://doi.org/10.1007/3-540-39568-7_20.
- [102] Jackson W.-A., Martin K. M. Cumulative Arrays and Geometric Secret Sharing Schemes. Advances in Cryptology — AUSCRYPT 1992. Lecture Notes in Computer Science. Berlin: Springer, 1993. Vol. 718. P. 48–55. DOI: https://doi.org/10.1007/3-540-57220-1_51.

- [103] Pfitzmann A., Hansen M. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. Version 0.34. 2010. Альтернативний запис: [TU Dresden record](#).
- [104] Chaum D. L. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*. 1981. Vol. 24, No. 2. P. 84–90. DOI: [10.1145/358549.358563](#).
- [105] Dingledine R., Mathewson N., Syverson P. Tor: The Second-Generation Onion Router. *Proceedings of the 13th USENIX Security Symposium*. 2004. [USENIX record](#).
- [106] Goldschlag D. M., Reed M. G., Syverson P. F. Onion Routing for Anonymous and Private Internet Connections. *Communications of the ACM*. 1999. Vol. 42, No. 2. P. 39–41. DOI: <https://doi.org/10.1145/293411.293443>.
- [107] Куперштейн Л. М., Кренцін М. Д. Аналіз тенденцій розвитку пірингових мереж. *Вісник Хмельницького національного університету. Технічні науки*. 2021. № 4(299). С. 25–29. DOI: 10.31891/2307-5732-2021-299-4-26-29.
- [108] Mittal P., Borisov N. ShadowWalker: Peer-to-peer anonymous communication using redundant structured topologies. *Proceedings of the 16th ACM Conference on Computer and Communications Security*. 2009. P. 161–172. DOI: 10.1145/1653662.1653683.
- [109] Freedman M. J., Morris R. Tarzan: A Peer-to-Peer Anonymizing Network Layer. *Proceedings of the 9th ACM Conference on Computer and Communications Security*. 2002. P. 193–206. DOI: <https://doi.org/10.1145/586110.586137>.
- [110] Lopes D., Dong J.-D., Castro D. F. S., Medeiros P., Barradas D., Portela B., Vinagre J., Ferreira B., Christin N., Santos N. Flow Correlation Attacks on Tor Onion Service Sessions with Sliding Subset Sum. *Proceedings 2024 Network and Distributed System Security Symposium*. San Diego, CA, USA, 26 February – 1 March 2024. DOI: 10.14722/ndss.2024.24337.
- [111] Куперштейн Л. М., Кренцін М. Д., Дудатьєв А. В., Каплун В. А. Аналіз проблем безпеки пірингових мереж. *Інформаційні технології та комп'ютерна інженерія*. 2022. Т. 54, № 2. С. 5–14. DOI: 10.31649/1999-9941-2022-54-2-5-14.
- [112] Pfitzmann A., Hansen M. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. Version 0.34. 2010.
- [113] Basyoni L., Fetais N., Erbad A., Mohamed A., Guizani M. Traffic Analysis Attacks on Tor: A Survey. *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIOT)*. Doha, Qatar, 2020. P. 183–188. DOI: 10.1109/ICIOT48696.2020.9089497.
- [114] Dingledine R., Mathewson N., Syverson P. Tor: The Second-Generation Onion Router. *Proceedings of the 13th USENIX Security Symposium*. San Diego, CA,

USA : USENIX Association, 2004. URL: <https://www.usenix.org/conference/13th-usenix-security-symposium/tor-second-generation-onion-router>.

- [115] Shirazi F., Simeonovski M., Asghar M. R., Backes M., Diaz C. A Survey on Routing in Anonymous Communication Protocols. ACM Computing Surveys. 2018. Vol. 51, No. 3. Article 51. P. 1–39. DOI: 10.1145/3182658.

ДОДАТКИ

ДОДАТОК А. Акти впровадження результатів дисертаційної роботи

АКТ

про впровадження результатів дисертаційної роботи
«Методи криптографічного захисту зображень»
аспіранта Вінницького національного технічного університету
Ціхоцького Микити Сергійовича

Результати дисертаційної роботи «Методи криптографічного захисту зображень» аспіранта Вінницького національного технічного університету Ціхоцького Микити Сергійовича, що полягають у розробці:

- методу шифрування зображень, який передбачає перестановку байтів у межах усього зображення та їх заміну;
- методу порогового розподілу секретного вмісту зображень за схемою (k, n) з використанням зашифрованих блоків і допоміжної складової дилера;
- програмного засобу, що реалізує запропоновані методи шифрування, розподілу та відновлення зображень,

впроваджені у систему захисту та розподіленого зберігання графічних даних ТОВ «Каскад-Безпека» для захищеного зберігання, передавання та відновлення зображень у розподіленому середовищі.

Директор ТОВ «Каскад-Безпека»
Коваль Артем Миколайович

«15» червня 2026 р.



ЗАТВЕРДЖУЮ

Проректор з науково-педагогічної роботи
та організації освітнього процесу
Вінницького національного технічного
університету

 Олександр ПЕТРОВ
«___» _____ 2026 р.

АКТ

про впровадження результатів дисертаційної роботи
ЦІХОЦЬКОГО МИКИТИ СЕРГІЙОВИЧА
за темою: «Методи криптографічного захисту зображень»,
поданої на здобуття ступеня доктора філософії
з галузі знань 12 «Інформаційні технології»
за спеціальністю 125 «Кібербезпека»

Комісія у складі декана факультету інформаційних технологій та комп'ютерної інженерії, к.пед.н., доцента Кирилашук Світлани Анатоліївни, к.т.н., доцента кафедри захисту інформації Войтович Олесі Петрівни та д.т.н., професора кафедри захисту інформації Лужецького Володимира Андрійовича склала цей акт про те, що на кафедрі захисту інформації Вінницького національного технічного університету у навчальний процес впроваджено такі результати дисертаційної роботи Ціхоцького М. С.:

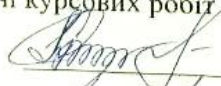
- метод шифрування зображень на основі побайтової підстановки та перестановки елементів у межах усього вектора даних;
- методи розподілу секретного вмісту зображень за схемами (n,n) та (k,n) з використанням попередньо зашифрованого зображення;
- протоколи застосування запропонованих схем для анонімної передачі даних зображень і розподіленого відмовостійкого зберігання зображень;
- програмні засоби, що реалізують процедури шифрування, розподілу та відновлення секретного вмісту зображень.

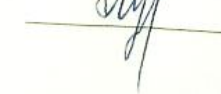
Зазначені результати використовуються під час проведення лекційних, лабораторних і практичних занять з дисциплін «Прикладна криптологія» та «Криптографічний захист інформації», а також при виконанні курсових робіт здобувачами вищої освіти.

Декан ФІТКІ, к.пед.н., доцент

к.т.н., доцент

д.т.н., професор





Світлана КИРИЛАШУК

Олеся ВОЙТОВИЧ

Володимир ЛУЖЕЦЬКИЙ

ТОВ «ІНТЕРНЕТ-РЕКЛАМА RIA»

ЄДРПОУ 37691230

IBAN: UA3630529900000260040061070325

АТ КБ "Приватбанк", МФО 305299



Юридична Адреса

22000, Україна, Вінницька обл., місто Козятин,
вулиця Незалежності, Будинок 39

Поштова Адреса

21015, Україна, м. Вінниця вул. А.С. Дігеля, 4 (4/б 4/)
тел./факс: +38 (0432) 503 100

АКТ

про використання результатів дисертаційної роботи

Ціхоцького Микити Сергійовича

«Методи криптографічного захисту зображень»

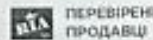
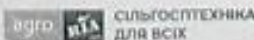
У розробках ТОВ «ІНТЕРНЕТ-РЕКЛАМА RIA», пов'язаних з побудовою систем захищеного передавання та відмовостійкого зберігання зображень використано такі результати дисертаційної роботи:

- Протокол передачі зображень, який використовує шифрування окремих частин зображення, що передаються різними маршрутами через проміжні вузли пірингової мережі. Це забезпечує анонімність користувачів і конфіденційність інформації.
- Протокол розподіленого відмовостійкого зберігання зображень, який передбачає зберігання окремих частин зашифрованого вмісту на різних запам'ятовувальних пристроях і відновлення зображення у разі наявності допустимої множини пристроїв, що відмовили. Це забезпечує підвищення гарантоздатності систем зберігання інформації.

Директор

ТОВ «ІНТЕРНЕТ-РЕКЛАМА RIA»

Ракуха Оксана Валентинівна



ДОДАТОК Б. Результати тестування програмної реалізації методу шифрування та розподілу зображень за (k,n) схемою

ПРОГРАМА ШИФРУВАННЯ ТА РОЗПОДІЛУ ЗОБРАЖЕННЯ

Метод: перестановка та підстановка байтів з подальшим розподілом за (k,n) -коаліційною схемою

[1] Зчитування вхідного зображення

Файл зображення: input_image_512x512.png

Тип зображення: 8-бітове напівтонове зображення

Розмір зображення: 512 x 512

Обсяг вхідних даних: 262144 байт

Обсяг вхідних даних: 0.250 MiB

Сформовано вхідний вектор V.

Довжина вектора V: 262144 байт

[2] Початкові параметри шифрування

Початкове значення генератора: 0xA1B2C3D4

Позиції зворотнього зв'язку: [0, 1, 21, 31]

Ключові біти для піддіапазонів: [0, 1, 0, 1]

Параметри підстановки:

$a = 65$, $b = 47$, $c = 111$

Ініціалізацію параметрів шифрування завершено.

Час ініціалізації: 0.000019 с

[3] Зашифрування зображення

Почато зашифрування вектора V.

Виконано перестановку елементів вектора.

Виконано підстановку значень байтів.

Сформовано зашифрований вектор C.

Довжина вектора C: 262144 байт

Час зашифрування: 0.028 с

Пропускна спроможність зашифрування: 9.30 МБ/с

Файл із зашифрованими даними збережено:

encrypted_image.bin

[4] Початкові параметри коаліційної схеми розподілу

Схема розподілу: (k,n) -коаліційна схема

Кількість учасників n : 5

Порогове значення k : 3

Кількість допустимих учасників для відновлення: не менше 3

Кількість блоків зашифрованого вектора: 5

Розмір одного блоку: 52429 байт

Позначення блоків:

V_1 , V_2 , V_3 , V_4 , V_5

Розмір дилерської частки D: 52429 байт

[5] Формування часток учасників

Почато розподіл зашифрованого вектора C.

Сформовано частку учасника P1:

склад частки: B1, B2, B3

розмір частки: 157287 байт

Сформовано частку учасника P2:

склад частки: B2, B3, B4

розмір частки: 157287 байт

Сформовано частку учасника P3:

склад частки: B3, B4, B5

розмір частки: 157287 байт

Сформовано частку учасника P4:

склад частки: B4, B5, B1

розмір частки: 157287 байт

Сформовано частку учасника P5:

склад частки: B5, B1, B2

розмір частки: 157287 байт

Сформовано дилерську частку D.

Час розподілу: 0.0019 с

Пропускна спроможність розподілу: 131.52 МБ/с

Файли часток збережено:

share_P1.bin, share_P2.bin, share_P3.bin, share_P4.bin, share_P5.bin, dealer_share_D.bin

[6] Перевірка можливості відновлення

Для перевірки обрано коаліцію учасників:

P1, P3, P5

Кількість наданих часток: 3

Порогове значення k: 3

Коаліція є допустимою.

Отримані блоки:

від P1: B1, B2, B3

від P3: B3, B4, B5

від P5: B5, B1, B2

Виявлено всі блоки, необхідні для відновлення.

Дилерську частку D підключено до процедури відновлення.

Почато відновлення зашифрованого вектора C.

Відновлення завершено.

Довжина відновленого вектора C': 262144 байт

Час відновлення: 0.0133 с

Пропускна спроможність відновлення: 18.66 МБ/с
 Відновлення зашифрованого вектора виконано коректно.

 [7] Розшифрування відновленого вектора

Почато розшифрування вектора С.

Виконано зворотну підстановку значень байтів.

Виконано зворотну перестановку елементів вектора.

Сформовано відновлений вектор V'.

Довжина вектора V': 262144 байт

Час розшифрування: 0.028 с

Пропускна спроможність розшифрування: 9.30 МБ/с

Кількість відмінних байтів: 0

Зображення відновлено без спотворень.

Файл відновленого зображення збережено:

restored_image_512x512.png

 [8] Підсумок виконання

Обсяг вхідного зображення: 262144 байт

Обсяг зашифрованого вектора: 262144 байт

Кількість учасників схеми: 5

Порогове значення: 3

Кількість сформованих часток учасників: 5

Кількість дилерських часток: 1

Час ініціалізації шифрування: 0.000019 с

Час зашифрування: 0.028 с

Час розподілу за коаліційною схемою: 0.0019 с

Час відновлення за коаліційною схемою: 0.0133 с

Час розшифрування: 0.028 с

Перевірка коректності розподілу: успішно

Перевірка коректності відновлення: успішно

Перевірка коректності розшифрування: успішно

Загальний результат:

вхідне зображення зашифровано, розподілено між учасниками,
 відновлено з допустимої коаліції та розшифровано без втрати даних.

=====

ЗАВЕРШЕНО

=====

ДОДАТОК В. Список публікацій за темою дисертації

1. Ціхоцький М.С., Лужецький В.А. Аналіз методів розподілу секрету Електронне фахове наукове видання "Кібербезпека: освіта, наука, техніка", вип. 1, вип. 25, с. 279–293, Вер 2024. DOI: <https://doi.org/10.28925/2663-4023.2024.25.279293>
2. Luzhetsky V., Tsikhotskyi M. Image encryption and distribution method based on LFSR and counters. Information Technologies and Computer Engineering, 22(3), 77-88, 2025. DOI: <https://doi.org/10.31649/vitce/3.2025.77>
3. Лужецький В.А., Ціхоцький М.С. Розподіл секретного вмісту даних за (k,n) -схемою з використанням зашифрованих блоків. Вісник Вінницького політехнічного інституту. №5, 113-120, 2025. DOI: <https://doi.org/10.31649/1997-9266-2025-182-5-113-120>
4. Лужецький В.А., Ціхоцький М.С. Схема порогового розподілення секрету « k,n » МНПК «ІТКМ-2024». 21 – 24 травень 2024.
URL: https://journal.compsc.if.ua/test/public/journals/zbirnyk_2024.pdf.
5. Лужецький В.А., Ціхоцький М.С. Протокол анонімної передачі зображень / Міжнародна науково-практична конференція "Інформаційні технології та комп'ютерне моделювання" (МНПК «ІТКМ-2024»). 227-228. 18 – 22 травень 2026. URL: <https://drive.google.com/file/d/14wz-rFrIYJyctAkV-lZ8RjykfPhT4Qqw/view>
6. Ціхоцький М.С. Метод розподілу зашифрованого зображення за (k,n) -схемою з використанням коаліцій учасників / Матеріали міжнародної науково-практичної інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2026)» Вінниця: ВНТУ, 2026. URL: <https://press.vntu.edu.ua/index.php/vntu/catalog/book/914>
7. Ціхоцький М. С. Протокол розподіленого зберігання медичних даних за пороговою (k,n) -схемою. ITSec: Безпека інформаційних технологій : матеріали XV Міжнар. наук.-техн. конф., м. Тернопіль, 27–29 трав. 2026 р. Тернопіль : ТНТУ ім. І. Пулюя, 2026. С. 145–146.