Міністерство освіти і науки України
Кам’янець-Подільський національний університет імені Івана Огієнка
Фізико-математичний факультет
Кафедра комп’ютерних наук

Дипломна робота
магістра

з теми: «ДОСЛІДЖЕННЯ ПЕРСПЕКТИВНИХ ПІДХОДІВ ДО КОДУВАННЯ МУЛЬТИМЕДІЙНИХ ДАНИХ»

Виконав: студент групи KN1-M22
спеціальності 122 Комп’ютерні науки
Коваль Олексій Олександрович

Керівник: **Смалько Олена Аркадіївна,**
доцент кафедри комп’ютерних наук,
кандидат педагогічних наук, доцент.

Рецензент: **Шелепало Галина Василівна,**
доцент кафедри захисту інформації Вінницького національного технічного університету,
кандидат фізико-математичних наук, доцент

Кам’янець-Подільський – 2023

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

КА – клітинний автомати (cellular automata).

КЛХК – кусково–лінійна хаотична карта.

ЛПКА – лінійна пам’ять клітинних автоматів (LMCA – linear memory cellular automata).

ТХ – теорія хаосу (theory of chaos).

ХС – хаотична система.

ВСТУП

Експоненціальне зростання мультимедійних даних за останні роки породило потребу в більш ефективних і безпечних способах кодування та обробки цих даних. Обсяг цифрового мультимедійного контенту станом на тепер продовжує динамічно зростати, збільшуються також проблеми стосовно забезпечення надійного зберігання величезної кількості інформації, ефективного її опрацювання та безпечного передавання мережами. Для вирішення цих проблем, зокрема, розробляються нові підходи до кодування мультимедійних даних. При цьому досить перспективними напрямками досліджень є використання клітинних автоматів і динамічних систем з хаотичною поведінкою. Дана кваліфікаційна робота присвячена дослідженню саме таких підходів.

Об'єктом дослідження є кодування мультимедійних даних.

Предметом дослідження є застосування клітинних автоматів та хаотичних систем для шифрування даних.

Метою кваліфікаційної роботи є дослідження, ґрунтовний аналіз та практична реалізація двох перспективних у наш час підходів до кодування мультимедійних даних, зокрема з використанням клітинних автоматів та динамічних систем, що демонструють хаотичну динаміку.

Завдання дослідження:

1) аналіз сучасної термінології та концепцій сфери шифрування даних;
2) дослідження перспективних методів та підходів до кодування мультимедійних даних, що спираються на використання динамічних хаотичних систем теорії хаосу та клітинних автоматів;
3) опис деяких підходів до шифрування цифрових даних за допомогою динамічних та хаотичних систем, клітинних автоматів і фрактальних функцій, які вбачаються найбільш релевантними;
4) оцінювання ефективності алгоритмів, що реалізовують описані підходи, та їх порівняльний аналіз за кількома ознаками;

5) виконання прикладного завдання по реалізації мовою Python алгоритму шифрування графічних даних з використанням системи нелінійних диференціальних рівнянь Росслера, кривої Гільберта та H-фракталу.

Впродовж виконання кваліфікаційної роботи використовувались наступні методи дослідження: аналіз літератури, тематичні дослідження, логіко-аналітичні та експериментальні дослідження, формалізація, моделювання, візуальні методи, аргументація та порівняння отриманих результатів.

Дана робота має теоретичне та практичне значення, зокрема одержані внаслідок її виконання результати дають змогу покращити свої знання у сфері криптографії. Зокрема надають глибоку та ґрунтовну інформацію, щодо використання хаотичних систем та клітинних автоматів для шифрування мультимедійних даних. Проведений аналіз та порівняння дозволяє наочно оцінити, використовувані методи та доцільність їх використання. Написаний код доцільно використовувати при вирішенні прикладних задач з шифрування.

Деякі отримані результати вдалося апробувати у вигляді статті *Метод використання динамічних перетворень для стиснення, захисту та приховування відеоінформаційних ресурсів в інфокомунікаційних системах. Сучасна спеціальна техніка* [1] (видання категорії **Б**), статті опублікованої на платформі IEEE – *Research of prospective encoding methods for multimedia content. 2022 IEEE 4th International Conference on Advanced Trends in Information Theory (ATIT)* (видання категорії **А**) [29], тез – *Дослідження перспективних підходів до кодування мультимедійних даних* у збірнику матеріалів наукової конференції здобувачів вищої освіти фізико-математичного факультету Кам’янець-Подільського національного університету імені Івана Огієнка [2].

Робота складається із робота складається із переліку умовних позначень, вступу, трьох розділів, висновків до розділів 1 та 2, загальних висновків, списку використаних джерел та додатку. У вступі описано актуальність теми, об’єкт, предмет, мета та завдання дослідження. У першому розділі подано аналіз сучасної

термінології та концепцій сфери шифрування даних. Другий розділ присвячений конкретним прикладам алгоритмів шифрування за допомогою клітинних автоматів та хаотичних систем. У третьому розділі ми розглянули власне алгоритми шифрування, прикладів, що були описані у розділі 2, детально проаналізували їх та провели порівняльний аналіз алгоритмів з хаотичною системою та клітинними автоматами. У висновках для розділів 1 та 2, підбито підсумки за кожним розділом. У загального висновку підведено підсумки, щодо досягнутої мети, виконаних завдань, практичного та теоретичного значень даного дослідження. У списку використаних джерел, зібрано інформацію, статті, публікації, журнали, що використовувались під час виконання роботи та публікації, за якими апробувалась робота. У додатку поданий програмний код на мові Python алгоритму шифрування на основі хаотичної системи Росслера та рисунок із результатом роботи алгоритму.

РОЗДІЛ 1.
ОСНОВИ КРИПТОГРАФІЇ ТА ПОШИРЕНІ ПІДХОДИ ДО КОДУВАННЯ ІНФОРМАЦІЇ

Сучасний світ неможливо уявити без передачі та обробки інформації. Завдяки технологічному прогресу та розширенню мережі Інтернет, обмін інформацією став надзвичайно важливою складовою нашого повсякденного життя. Проте разом із зростанням обсягів інформації зростають і загрози її конфіденційності та цілісності.

Криптографія – це наука, що вивчає методи та засоби захисту інформації від несанкціонованого доступу та зміни. Вона відіграє важливу роль в сферах як інформаційної безпеки, так і захисту приватності особистих даних. Історія криптографії налічує тисячоліття розвитку, і сьогодні ця галузь має важливий вплив на багато аспектів нашого життя, включаючи фінанси, комунікації та технології. Цей розділ присвячений дослідженню основ криптографії та поширеним підходам до кодування інформації. Нижче ми представимо фундаментальні концепції криптографії, розглянути різні методи та алгоритми шифрування.

1.1. Основні поняття теорії кодування та криптографії

Кодування (англ.coding) – процес перетворення великих обсягів мультимедійної інформації, такої як відео, аудіо та зображення, в бінарний формат (нулі та одиниці).

Вихідне кодування (англ. source coding) у контексті передавання і захисту даних називають кодування, зроблене в джерелі даних перед тим, як вони будуть збережені або передані.

Шифрування [кодування] (англ.encoding) – процес перетворення даних у такий спосіб, щоб лише особи з відповідним ключем могли їх дешифрувати та прочитати.

Дешифрування – процес перетворення зашифрованих даних в текст зрозумілий для читання.

Блокове шифрування – це метод шифрування, при якому вхідні дані (текст або дані будь-якого формату) розбиваються на блоки фіксованого розміру, і кожен блок обробляється окремо. Кожен блок зазвичай шифрується індивідуально, використовуючи один і той же ключ шифрування. [4].

Потокове шифрування – система підстановки на алфавіті блоків(елементів визначеної довжини).

Криптографічні хеш–функції – криптографіче перетворення двійкової послідовності довільної довжини у двійкову послідовність фіксованої довжини

Динамічні перетворення – концепція в галузі криптографії, яка стосується змінності ключів та алгоритмів шифрування з часом. У традиційних схемах шифрування статичний ключ використовується для захисту інформації. У динамічних перетвореннях, ключі та алгоритми можуть змінюватися відповідно до певних правил та параметрів, що дозволяє покращити безпеку інформації.

Криптографічний алгоритм, або шифр – це математична формула, що описує процеси шифрування і розшифрування.

Плутанина – одна з ключових концепцій в криптографії, яка полягає в тому, щоб зробити зв'язок між вхідними даними і вихідним шифром як складним і непередбачуваним, наскільки це можливо. Ідея полягає в тому, щоб змішувати вхідні дані з ключем шифрування так, щоб будь–яка зміна в одному біті вхідних даних або ключа впливала на всі біти в шифротексті.

Дифузія – техніка заміни значення пікселя зображення, зміна значення пікселя вплине на зміну в інших значеннях пікселів так, як пікселі в різних позиціях з’єднані. Інакше кажучи, кожен біт вихідного шифру повинен залежати від всіх бітів вхідних даних та ключа. Ця властивість гарантує, що навіть при малій зміні в вхідних даних шифротекст буде значно відрізнятися.

Скремблювання – шифрування потоку даних, в результаті якого він виглядає як потік випадкових бітів.

Випадкова величина – величина, можливими значеннями якої є результати випробувань чи спостережень явищ або процесів, що мають випадковий характер

Псевдовипадкові послідовності (числа) – послідовності, що отримуються за цілком невипадковим алгоритмом, але мають властивості, дуже подібні до властивостей реалізацій випадкових чисел.

Хаотичні системи [ХС] (за Робертом Девані) – це системи, які:

1. мають чутливу залежність від початкових умов;
2. є топологічно транзитивним (тобто, для будь-яких двох відкритих множин деякі точки з однієї множини зрештою потраплять в іншу множину);
3. їхні періодичні орбіти утворюють щільну множину.

Показник Ляпунова динамічної системи описує швидкість, з якою дві (близькі одна до одної) точки фазового простору віддаляються або наближаються одна до одної (залежно від знака). Додатність показника Ляпунова зазвичай свідчить про хаотичну поведінку системи. Потік динамічної системи визначають як однопараметричне сімейство відображень: $F^t(x_0) = x_t$, де $t \mapsto x_t$ позначає траєкторію в динамічній системі. Показник Ляпунова визначається:

$$\lambda(x) = \lim_{t\to\infty} \frac{1}{t} \cdot \ln \|d_x F^t\|$$

Атрактор – множина точок у фазовому просторі, до якої збігаються фазові траєкторії дисипативної системи.

Дивний (хаотичний) атрактор – математичний образ детермінованого хаосу у фазовому просторі, який відображає складну та непередбачувану динаміку системи. Він є графічним відображенням траєкторії системи в динаміці, де система може перебувати у стані хаосу [5].

Клітинні автомати [англ. cellular automata, CA] (КА) – це обчислювальна модель, яка моделює поведінку складних систем за допомогою простих правил і локальних взаємодій.

Кубіт (від англ. «**qu**antum **bit**») – це фундаментальна одиниця квантової інформації в квантовому обчисленні. Квантовий комп'ютер використовує кубіти

для обробки та зберігання інформації у квантовій формі, що відрізняється від класичних бітів, які використовуються в звичайних комп'ютерах.

1.2. Різноманіття існуючих підходів до кодування цифрових даних

З розвитком цифрової епохи важливість збереження та передачі мультимедійних даних стала критичною. Кодування мультимедіа стали необхідними компонентами для забезпечення конфіденційності та цілісності цих даних.

Ми розглянемо широкий спектр методів і підходів до кодування та шифрування мультимедіа, включаючи класичні, сучасні та перспективні методи. Це дозволить нам краще зрозуміти різноманіття та розвиток цих методів і їхню важливість у сучасному цифровому світі [4].

1.2.1. Завдання кодування та традиційні напрями в шифруванні

Основні напрями теорії кодування

Кодування мультимедійних даних є фундаментальним процесом, який передбачає перетворення необроблених мультимедійних даних у формат, який можна ефективно зберігати, передавати та обробляти. Щоб досягти цього, необхідно враховувати кілька основних завдань [4]

Одним із завданням кодування мультимедійних даних є стиснення.

Стиснення – процес зменшення розміру мультимедійних даних без втрати значної інформації. Це досягається видаленням надмірностей у даних, які можуть бути часовими або просторовими. Часова надлишковість стосується того факту, що мультимедійні дані містять багато повторюваної інформації протягом певного часу, такої як кадри у відеоряді або ноти в музичному творі. Просторова надмірність стосується того факту, що мультимедійні дані містять багато повторюваної інформації в одному кадрі, такої як пікселі в зображенні або частоти в аудіо сигналі.

Іншим завданням кодування мультимедійних даних є забезпечення стійкості до помилок. Стійкість до помилок означає здатність мультимедійних даних відновлюватися після помилок передачі або втрат без значного погіршення якості. Це особливо важливо в мультимедійних програмах, де втрата даних може призвести до помітного погіршення якості, як–от відеоконференції або потокове передавання.

Третім завданням кодування мультимедійних даних є масштабованість. Масштабованість означає здатність мультимедійних даних ефективно кодуватися з різними бітрейтами (швидкість проходження бітів інформації за секунду), роздільними здатностями або рівнями якості. Це важливо в мультимедійних програмах, де пристрої кінцевих користувачів мають різні можливості або умови мережі, наприклад потокове відео на смартфони чи телевізори [31].

Нарешті, безпека є важливим завданням кодування мультимедійних даних. Мультимедійні дані можуть містити конфіденційну інформацію, яку потрібно захистити від несанкціонованого доступу або модифікації. Цього можна досягти за допомогою методів шифрування [20] та цифрових водяних знаків [4].

Симетричне та асиметричне шифрування

Симетричне шифрування та асиметричне шифрування – це два основних методи шифрування, які використовуються для захисту мультимедійних даних та інших видів інформації.

Симетричне шифрування – використовує один і той самий ключ для як шифрування, так і розшифрування даних. Відправник і отримувач беруть цей ключ і використовують його для перетворення звичайного тексту в шифрований та навпаки. Однак важливо, щоб ключ залишався в таємниці від невповноважених осіб, оскільки він може бути використаний для розшифрування даних.

За допомогою даного методу шифрування реалізовано такі відомі алгоритми як: DES (Data Encryption Standard), AES (Advanced Encryption Standard) і IDEA (International Data Encryption Algorithm).

Асиметричне шифрування використовує два ключі – відкритий і секретний. Відкритий ключ використовується для шифрування даних, тоді як секретний

ключ використовується для розшифрування. Основна властивість полягає в тому, що інформація, зашифрована одним ключем, може бути розшифрована тільки іншим ключем.

Асиметричне шифрування використовують: RSA (Rivest–Shamir–Adleman), ECC (Elliptic Curve Cryptography) і DSA (Digital Signature Algorithm).

Перестановки

Метод перестановки – один із основних методів шифрування, який використовується для забезпечення безпеки і конфіденційності інформації. Цей метод полягає в перетворенні символів або блоків символів у шифр шляхом їх перестановки в певному порядку.

Метод перестановки базується на ідеї перетворення символів або блоків символів шляхом їх перестановки згідно з певним ключем. Зазвичай цей ключ визначає, яким чином символи повинні бути переставлені. Переставлення може відбуватися по рядках, стовпцях або іншим чином. Наприклад, в алгоритмі шифру "перестановка по стовпцях" символи повідомлення розміщаються в матриці, і символи переставляються в порядку, вказаному ключем, який може бути перестановкою чисел, букв або символів.

Підстановки

Підстановки метод шифрування, при якому символи відкритого тексту замінюються на інші символи або символьні групи у шифртексті. Це може бути здійснено за допомогою різних механізмів, таких як заміна літер, перестановка символів, чи інші алгоритми заміни. Наприклад, у простій підстановці літер, кожна літера відкритого тексту замінюється на певну іншу літеру у шифртексті.

Існують доволі складні методи підстановок, такі як шифр Віженера чи шифр Плейфера, які використовують ключі для керування процесом шифрування. Підстановки використовуються у шифруванні, але деякі методи, зокрема прості методи підстановок, легко можуть бути розкриті шляхом криптоаналізу. У сучасних криптографічних системах використовують більш складні методи, які стійкі до різних атак, таких як атаки з використанням статистичних підходів та методи криптоаналізу.

Теорія груп

Теорія груп — це розділ математики, який вивчає симетрію та структуру через поняття групи. Група — це набір елементів разом із операцією, яка об’єднує будь-які два елементи для утворення третього елемента, що задовольняє певні властивості.

У криптографії теорія груп використовується для забезпечення безпечного зв’язку, гарантуючи, що інформація залишається конфіденційною, навіть якщо її перехопить неавторизована сторона. Одним із поширених застосувань теорії груп у криптографії є використання її для створення асиметричних алгоритмів шифрування, таких як алгоритм RSA.

Теорія груп також використовується в розробці криптографічних хеш-функцій, які використовуються для надання унікальних вихідних даних фіксованого розміру для будь-якого заданого введення, надаючи засоби перевірки цілісності даних. Безпека багатьох криптографічних протоколів і систем залежить від властивостей груп, таких як складність задачі дискретного логарифмування, яка передбачає знаходження експоненти заданого числа в скінченній циклічній групі.

Загалом, теорія груп забезпечує фундаментальну основу для розуміння властивостей безпеки багатьох криптографічних систем і протоколів, а також є важливим інструментом для проектування та аналізу захищених систем зв’язку[33].

Блочне шифрування

Блокові шифри що є основою, на якій реалізовані практично всі криптосистеми. Методика створення ланцюжків із зашифрованих блочними алгоритмами байт дозволяє шифрувати ними пакети інформації необмеженої довжини. Таку властивість блокових шифрів, як швидкість роботи, використовується асиметричними криптоалгоритмами, повільними за своєю природою. Відсутність статистичної кореляції між бітами вихідного потоку блочного шифру використовується для обчислення контрольних сум пакетів даних і в хешуванні паролів

Стеганографія

Стеганографія – це процес приховування інформацію в інших даних так, щоб ця інформація залишалася непомітною для невповноважених користувачів. Цей метод використовується для приховування інформації у мультимедійних даних, таких як зображення, аудіо та відео. Стеганографія була використана ще в давньому Єгипті, де інформація приховувалася на папірусах та інших документах.

Стеганографія в мультимедійних даних використовує методи для вбудовування інформації в малі зміни в даних. Основний принцип полягає в тому, що інформація кодується шляхом невеликих змін в кольорах, аудіо сигналах або пікселях мультимедійних файлів. Ці зміни зазвичай настільки малі, що їх не помічає людське око або вухо, але можуть бути відновлені за допомогою відповідного програмного забезпечення.

1.2.2. Перспективні підходи до шифрування

В пошуках ефективних способів шифрування цифрових даних криптографи проявляють високу майстерність та виключну винахідливість, шукаючи та використовуючи в своїх розробках оригінальні математичні структури, нетривіальні функції та певні релевантні теорії. Так, зокрема, в алгоритмах шифрування з’явилися фрактали, трикутник Паскаля, криві Піано, Гільберта, КА та ХС.

Квазігрупи

На відміну від теорії груп, яка є більш широкою математичною областю, квазігрупи є конкретною алгебраїчною структурою, яка може бути використана для шифрування. Квазігрупа є природним узагальненням поняття групи, їхня відмінність від груп полягає в тому, що вони не повинні бути асоціативними[27].

Квазігрупи стали предметом досліджень у сфері криптографії, пропонуючи альтернативний підхід до створення безпеки у криптографічних системах.

Основні ідеї, які використовуються для шифрування та кодування на основі квазігруп, включає використання операції композиції в квазігрупі, що може бути використана для виконання шифрування, створення кодових просторів, кодування та декодування повідомлень. Замість того, щоб використовувати стандартні математичні структури, такі як групи чи поля, можна використовувати квазігрупові властивості для створення шифру. Саме операції композиції, які визначають внутрішню структуру множини, дозволяють ефективне використання математичних властивостей теорії груп для шифрування.

Хоча є широкі можливості для використання квазігруп у криптографії, важливо враховувати, що такий підхід може стикатися із складнощами щодо ефективності та стійкості порівняно з іншими сучасними криптографічними методами, безпека криптосистеми, яка базується на квазігрупах, вимагає ретельного математичного аналізу та перевірки властивостей квазігруп.

Еліптичні криві

Даний підхід, заснований на алгебраїчній структурі еліптичних кривих над скінченними полями, такі криві складаються з точок, які задовільняють рівняння $y^2 = x^3 + ax + b$. Шифрування на еліптичних кривих дозволяє використовувати менші ключі порівняно з іншими підходам для забезпечення еквівалентної безпеки, а операції над еліптичних кривих вимагають менше обчислювальних ресурсів порівняно з іншими математичними областями.

Еліптичні криві застосовуються для обміну ключами, цифрових підписів, псевдовипадкових генераторів тощо[36].

Клітинні автомати

КА – це обчислювальна модель, яка моделює поведінку складних систем за допомогою простих правил і локальних взаємодій. Його широко застосовують у різних сферах, включаючи обробку зображень, стиснення даних, криптографію та шифрування. У цьому розділі ми дослідимо використання КА у шифруванні, зосередивши увагу на його основних принципах і застосуваннях

Основним принципом шифрування за допомогою КА є перетворення відкритого тексту в зашифрований текст за допомогою ряду математичних

операцій на основі правила КА. Відкритий текст розділений на клітинки, і кожній клітинці призначається унікальний стан відповідно до правила шифрування. Процес шифрування включає генерацію серії ключових послідовностей, які використовуються для зміни стану комірок. Остаточний зашифрований текст отримується шляхом вилучення змінених станів клітинок і перетворення їх у двійкову послідовність.

КА показали перспективу в шифруванні мультимедійних даних, таких як зображення та відео. Просторові властивості КА можна використовувати для кодування пікселів або блоків зображення.

Однією з переваг використання КА для шифрування є їх здатність генерувати складні та непередбачувані шаблони з простих правил. Вибравши відповідні правила КА та початкові умови, процес шифрування може створити високий ступінь випадковості та непередбачуваності, що підвищує безпеку шифрування. Крім того, автоматизоване КА шифрування стійке до диференціальних атак та інших методів криптоаналізу, що робить його надійним методом шифрування.

Існує кілька підходів до використання КА для шифрування, включаючи потоковий шифр і блоковий шифр. У потоковому шифруванні відкритий текст шифрується по одному біту за допомогою псевдовипадкової послідовності ключів, створеної за правилом КА. Блоковий шифр шифрує відкритий текст у вигляді блоків фіксованого розміру за допомогою правила КА, що залежить від ключа. Вибір типу шифру залежить від конкретних вимог та задач. Такими можуть бути розмір даних, які потрібно зашифрувати, швидкість і бажаний рівень безпеки.

Іншим важливим аспектом шифрування КА є керування ключами. Безпека шифрування залежить від секретності та випадковості послідовностей ключів. Тому важливо розробити ефективні та безпечні методи генерації та розповсюдження ключів, щоб забезпечити конфіденційність шифрування.

Теорія хаосу

Теорія хаосу (ТХ) широко вивчалася в галузі криптографії, як засіб забезпечення безпечного зв’язку та шифрування даних. Нижче ми оглянемо основні принципи ТХ та її застосування в криптографії.

ТХ займається вивченням нелінійних систем, які демонструють складну та непередбачувану поведінку. Ці системи характеризуються своєю чутливістю до початкових умов, що означає, що невеликі зміни в початкових умовах можуть призвести до дуже різких змін в кінцевих результатах. Ця властивість зробила теорію хаосу особливо цікавою для використання в криптографії, де її можна використовувати для створення безпечних алгоритмів шифрування [22], [26].

Основним принципом криптографії на основі ТХ є використання хаотичної поведінки системи для створення випадкового ключа, який використовується для шифрування повідомлення. Ключ генерується шляхом вибору початкових умов для хаотичної системи та дозволу їй розвиватися з часом. Отримана траєкторія системи використовується як ключ для шифрування.

Однією з головних переваг криптографії на основі хаосу є її стійкість до атак на основі математичних алгоритмів. Традиційні методи шифрування ґрунтуються на використанні математичних алгоритмів, які можуть бути використані зловмисниками. Хаотична поведінка системи, яка використовується, ускладнює завдання зловмисникам відновити ключ, який використовується для шифрування [21].

Використовуючи хаотичні карти, атрактори або інші хаотичні системи в процес шифрування, стає можливим створювати алгоритми шифрування з підвищеною безпекою. Складна динаміка ХС створює високий ступінь випадковості та ускладнює зловмисникам розшифровку зашифрованих даних без належних ключів [17] .

Також ХС може забезпечувати синхронізацію та безпечний зв'язок. Синхронізація хаосу (за якої дві або більше хаотичних систем можуть досягти однакової поведінки), ще один аспект ХС, знайшла застосування в безпечному обміні даними. Хаотичні системи можуть бути синхронізовані між відправником і одержувачем для забезпечення безпечної передачі інформації. Використовуючи

властивості синхронізації хаотичних систем, можна розробити методи шифрування, щоб забезпечити безпечні та надійні канали зв’язку [15].

Однак існують також проблеми, пов'язані з використанням ТХ в криптографії. Однією з головних проблем є необхідність гарантувати, що система, яка використовується для шифрування, є справді хаотичною, оскільки незначні відхилення від хаосу можуть призвести до втрати безпеки. Крім того, криптографія, заснована на хаосі, може потребувати об’ємних обчислень, що може обмежити її практичне застосування [32].

Незважаючи на ці проблеми, останніми роками було проведено численні дослідження та дослідницькі зусилля для вивчення потенціалу ХС в криптографії. Наприклад, було досліджено використання хаотичних карт і систем, таких як системи Лоренца, Чуа та Ресслера, для цілей шифрування. Також досліджували використання гібридних методів шифрування, які поєднують хаотичні системи з іншими методами шифрування, такими як Advanced Encryption Standard (AES) [9].

Поєднання клітинних автоматів і динамічних систем із хаотичною поведінкою

Застосування КА і ХС в криптографії та їх поєднання дають можливість для нових досліджень у цій сфері.

Одним із основних застосувань КА і ХС в криптографії є генерація ключів. Властиву випадковість і чутливість до початкових умов хаотичних систем у поєднанні з еволюцією КА на основі правил можна використовувати для генерації криптографічних ключів. Ці ключі стійкі до атак і забезпечують високий рівень безпеки для алгоритмів шифруваннях [22], [24].

Хаотичні клітинні автомати. Поєднуючи поняття КА і ХС, хаотичні КА виникли як сучасний підхід у криптографії. Хаотичні клітинні автомати використовують чутливу залежність від початкових умов і складну поведінку хаотичних систем для створення випадкових і непередбачуваних послідовностей для криптографічних цілей [17].

Станом на сьогодні підхід до інтеграції КА і ХС в криптографії продовжує розвиватися в міру того, як дослідники все глибше вивчають властивості та застосування цих концепцій. Поточні дослідження зосереджені на оптимізації

алгоритмів шифрування, аналізі їх властивостей безпеки, вивченні нових хаотичних карт або атракторів і розробці практичних реалізацій для безпечного зв’язку та захисту даних.

Хеш–функції

Хеш–функції – важливий інструмент криптографії, який використовується для генерації фіксованого розміру вихідних даних (хешу) з довільного об'єкта даних вхідного розміру. Хеш–функції мають декілька основних властивостей, які роблять їх корисними в різних аспектах криптографії:

1. Фіксований розмір вихідних даних – хеш–функції завжди генерують вихідні дані фіксованого розміру, незалежно від розміру вхідних даних. Це означає, що навіть якщо вхідні дані дуже великі або малі, хеш завжди буде одного розміру.
2. Велика різноманітність вхідних даних – навіть найменші зміни в початкових даних повинні призводити до значних змін у вихідному хеші. Ця властивість робить хеш–функції відмінними для виявлення навіть найдрібніших змін в даних.
3. Неверифікована обчислювальна складність – обчислення хешу з вихідними даними дуже швидко і легко, але обернене обчислення – визначення вихідних даних з хешу – повинно бути обчислювально складним завданням.

Хеш–функцій знайшли такі застосування в криптографії:

1. Збереження паролів – хеш–функції використовуються для збереження паролів в безпеці. Замість зберігання фактичного паролю, система зберігає лише його хеш. При введенні пароля система порівнює хеш введеного пароля із збереженим. Це захищає паролі від розголошення у випадку витоку даних.
2. Перевірка цілісності даних – хеш–функції використовуються для перевірки цілісності даних. Користувач може обчислити хеш даних та порівняти його з вихідним, щоб переконатися, що дані не були змінені під час передачі.

3. Блокчейн та криптовалюти – в криптовалютних системах, таких як Біткойн, хеш–функції використовуються для створення блоків даних, а також для забезпечення безпеки транзакцій та інших операцій в мережі.
4. Цифровий підпис – хеш–функції грають важливу роль у створенні цифрових підписів, які використовуються для підтвердження автентичності та цілісності даних.

Хеш–функції важливі для безпеки та цілісності даних в криптографії та знаходять застосування в різних аспектах цієї галузі. Вони допомагають захищати дані від несанкціонованого доступу та забезпечувати їх цілісність у віртуальному світі

Теорія граток

Теорія граток виникла у криптографії, як відповідь на розвиток квантових обчислювальних технологій. Оскільки існує можливість, що квантовий комп'ютер зможе зламати існуючі криптографічні алгоритми, тому виникла необхідність розробити нові стандарти для криптографії, які залишалися б безпечними за наявності квантових комп'ютерів.

Теоріїя граток, використовує математичні структури, які виглядають як гратки чи мережі точок, розташовані в просторі. Це може бути уявлено як сітка у просторі, де кожна точка має цілочисельні координати. Гратки можуть бути визначені у різних просторах та математичних структурах.

Основна ідея полягає у використанні проблеми решітки, де необхідно знайти короткий вектор в гратці.

Найпоширеніша версія цієї задачі - це "Задача короткої базисної решітки", де метою є знайти короткий ненульовий вектор в гратці. Знайти найкоротший вектор у гратці може бути ресурсо затратним завданням тому, що при великих розмірах гратки і великій кількості точок у гратці важко визначити, який саме вектор є найкоротшим

Квантова криптографія

Квантова криптографія – це спеціалізована галузь криптографії, яка використовує принципи квантової механіки для захисту інформації від

несанкціонованого доступу та забезпечення конфіденційності та надійності обміну даними. Квантова криптографія виходить за рамки традиційних методів криптографії та спирається на квантові властивості частинок для забезпечення безпеки [3].

Ідея квантової криптографії виникла в середині 20–го століття. Один із піонерів цієї галузі, Стівен Вінтерс, запропонував концепцію квантового розподілення ключа в 1968 році. Згодом, в 1984 році, Чарльз Беннетт та Ґільберт Брассар створили принцип квантової криптографії, відомий як протокол BB84.

Квантова криптографія базується на кількох квантових явищах:

1. Положення не визначене заздалегідь: за принципами квантової механіки, визначення стану квантової системи може бути виконане лише в момент вимірювання. Це означає, що не можна передбачити стан квантових бітів (кубітів) до їх вимірювання.
2. Паралельність: квантові системи здатні знаходитися в багатьох станах одночасно, що робить їхню обробку величезною швидкою та потужною.
3. Співположеність: стани квантових систем можуть бути співположеними, що дозволяє одночасно кодувати більше інформації.

Квантова криптографія використовує ці квантові явища для створення безпечних комунікаційних каналів та розподілу квантових ключів. Основні концепції включають:

1. Квантовий розподіл ключа: два абоненти створюють пару співположених кубітів і вимірюють їх стани. Будь–яке намагання спостерігача змінити квантовий стан приведе до виявлення змін в системі, що застосовується для виявлення злому або перехоплення ключа.
2. Квантова телепортація: кубіти можуть бути транспортовані через великі відстані, зберігаючи їхні стани. Це може бути використано для дистанційного розподілу ключів.

Висновок розділ 1

В цьому розділі було розглянуті поняття, що є необхідними для розуміння наступних розділів, описали, як вже відомі так і перспективні методи та підходи до шифрування даних. Одними з таких методів стали КА і ХС, що використовуються, як сучасні підходи в криптографії для підвищення безпеки даних і шифрування, пропонують унікальні перспективи та методи генерації безпечних криптографічних ключів і розробки алгоритмів шифрування, саме вони є основною темою майбутніх розділів.

КА і ХС, які відомі своєю високою складністю і непередбачуваністю, що може зробити криптографічні методи, засновані на них, дуже стійкими до атак інженерного методу та атак грубою силою, виправдовують, їх використання в криптографії. Відсутність статистичних закономірностей є дуже важливою якістю, оскільки такі закономірності можуть використовуватися для криптоаналізу. КА і ХС застосовуються у криптосистемах для важко передбачуваного розподілу даних, що робить їх ефективними для захисту інформації.

Також у розділі було розглянуті інші перспективні підходи до криптографічного захисту інформації, зокрема використання квазігруп, теорії граток, еліптичних кривих, теорії граток, хеш-функцій та квантової криптографії.

РОЗДІЛ 2.
ПЕРСПЕКТИВНІ ПІДХОДИ ДО ШИФРУВАННЯ КОМПОНЕНТІВ МУЛЬТИМЕДІА

Нижче буде описано підходи шифрування з використанням фрактальних функцій, ХС та КА, де математика переплітається з криптографією, відкриваючи цим шлях до нових досліджень та застосувань.

2.1. Шифрування зображень на основі хаотичної системи

Серед ефективних підходів до шифрування цифрових даних дедалі популярнішими стають ті, що ґрунтуються на теорії детермінованого хаосу. Як відомо, засновником сучасної ТХ, яка зосереджується на поведінці динамічних систем, що дуже чутливі до початкових умов, є американський математик і метеоролог Едвард Лоренц. Його концепція «Ефекту метелика» або основний принцип хаосу сформувався наприкінці 1950-х років у результаті невдалих спроб чисельного прогнозування погоди за допомогою лінійних статистичних моделей та через помилки округлення в деяких обчисленнях.

ТХ має справу з системами, що еволюціонують у часі до певного типу динамічної поведінки [30]. Такі системи еволюціонують за певними законами, а хаос має місце не у усіх детермінованих нелінійних системах. Власне про хаос, можемо говорити, коли існує стійкий і безладний довгостроковий розвиток системи, при цьому мають виконуватись певні математичні умови, а саме:

1. Динамічна нестабільність, що також називається ефектом метелика, це властивість чутливості до початкових умов, де початкові значення, можуть розвиватися зі значними відмінностями [16].
2. Топологічне змішування – властивість системи еволюціонувати з часом так, що будь-яка дана область станів завжди перекривається з будь-якою іншою даною областю [23].
3. Аперіодичність – система розвивається за орбітою, яка ніколи не повторюється, тобто ці орбіти ніколи не є періодичними.

4. Щільні періодичні орбіти – означає , що система слідує динаміці, яка може як завгодно близько наближатися до кожного можливого асимптотичного стану.
5. Ергодичність – статистичні виміри змінних дають подібні результати як у часі, так і у просторі.
6. Само подібність – еволюція системи в часі чи просторі має однаковий вигляд на різних масштабах спостереження. Ця характеристика призводить до враження, що система само повторюється на різних масштабах спостереження[11],[34].

Якщо кілька різних початкових хаотичних умов розвиваються у фазовому просторі таким чином, що ніколи не повторюються, тоді хаос є непередбачуваним. Але для розв'язання задач шифрування цілком досить псевдохаотичного характеру функцій.

Деякі науковці, наприклад фізик–математик Георгій Заславський, уродженець Одеси, стверджує, що псевдохаотичні функції мають нульові показники Ляпунова. Існує інша думка, що псевдохаос характеризується дуже малими показниками Ляпунова. Отож саме такі функції цілком доцільно використовувати для шифрування контенту.

Псевдохаотичні системи можуть мати нульові чи дуже малі додатні показники Ляпунова, навіть тоді, коли вони проявляють непередбачувану або хаотичну поведінку.

Функції з дуже малими додатними показниками Ляпунова можуть вказувати на слабку чутливість системи до початкових умов та можуть мати різні характеристики. Нижче наведено деякі приклади системи, для яких можливі дуже малі показники Ляпунова:

1. Стійкі атрактори – це такі системи мають дуже малі показники Ляпунова, оскільки траєкторії системи можуть повторюватись з часом, а сама система матиме малу чутливість до вихідних умов.
2. Квазіперіодичні системи – це системи, які мають квазіперіодичну динаміку, характеризуються дуже малі показниками Ляпунова, оскільки їх траєкторії є

вкрай чутливими до початкових умов, але при цьому можуть проходити навколо обмежених просторових ділянок.

3. Фрактальні атрактори, які мають дуже малі показники Ляпунова, – це фрактальні системи, що визначаються досить складною та непередбачуваною динамікою, і можуть демонструвати обмежену чутливість до початкових умов.
4. Системи зі стійкими стаціонарними точками – це системи з дуже малими показниками Ляпунова, що мають окремі стаціонарні точки.
5. Системи з критичними точками – це системи з дуже малими показниками Ляпунова, що проявляють феномен критичної поведінки в перехідних процесах і зберігають свою динаміку, при цьому в деяких околах можуть бути схильною до динамічних змін.
6. Коливальні системи зі зменшенням амплітуди - це системи з дуже малими показниками Ляпунова в яких амплітуда коливань зменшується з часом. Це може вказувати на довгострокову стійкість.

Втім все ж надійніше реалізовувати захист з використанням фрактальних структур.

Особливості детермінованих хаотичних систем, які за своєю суттю є непередбачуваними протягом тривалих періодів часу, відкривають можливості широкого їх застосування, наприклад, в якості одного з алгоритмів для побудови комплексного захисту цифрових ресурсів в інфокомунікаційних системах. Зокрема, динамічні системи з хаотичною або з псевдохаотичною поведінкою цілком підходять для шифрування різного цифрового контенту, особливо мультимедійного. Лише важливо пам’ятати, що для додаткової надійності функції повинні мати набір додатних показників Ляпунова.

Також подеколи варто їх ускладнювати, додаючи до них турбулентності за допомогою хоча б однієї функцій збурення.

Наведемо алгоритм шифрування зображень на основі властивості заповнення простору кривої Гільберта та властивості нескінченності Н–фрактала, яка поєднує в собі псевдовипадковість гіперхаотичної системи [18]. Даний метод

дасть нам змогу розібратись, як такий підхід, як ТХ поєднується з математикою для рішення завдань криптографії.

Італійський математик Піано та німецький математик Гільберт відкрили криву FASS (англ. space–Filling, self–Avoiding, Simple, and Self–similar curves), що заповнює квадратну сітку у 1890 і 1891 роках відповідно і дали метод для обходу кожного вузла в сітці за допомогою цієї безперервної кривої, яку називають кривою Гільберта. Крива Гільберта є кривою FASS, тобто кривою, що заповнює простір, само уникаючою, само подібною та простою кривою. Ці криві розташовані в Евклідовому просторі з розмірністю більше 1 і мають непорожні інтер'єри в просторі. На рисунку 2.1 показано такий шлях щоб крива Гільберта пройшла сітку.

Криву Гільберта першого порядку можна описати так: поділіть квадрат на чотири маленькі квадрати, починаючи з центру квадрата в нижньому лівому куті, потім прямуючи до центр квадрата у верхньому лівому куті, потім продовжуючи до центру квадрата у верхньому правому куті та потім вниз праворуч. Після прибуття в центр квадрата в нижньому куті, це завершує перший ітерації, а результат показано на рисунку. 2.2(2). Позиції початкової та кінцевої точок кривої Гільберта визначають його напрямок. В образі воно визначає порядок в якому він перетинає просторові пікселі. Тому, згідно до вибору та поєднання початкової та кінцевої точок кривої Гільберта, вісім різних схем скремблювання можуть генерується для кривої Гільберта першого порядку, як показано на рис 2.2 (2) розміщено у верхньому лівому та правому верхньому кутах кути сітки 2×2, показаної на рисунку 3(a). Два зображення фігури 2(b) повернуто на 90 градусів за годинниковою стрілкою і 90 градусів проти годинникової стрілки, відповідно, а потім розміщено у нижньому лівому та нижньому правому кутах відповідно. Другу криву можна отримати шляхом підключення суміжної кінцеві точки кривої, як показано на рисунку 3(a). На рисунку 3(b) показана крива Гільберта після трьох ітерацій. Повторюючи описані вище операції, а можна отримати криву, яка перетинає всю квадратну сітку

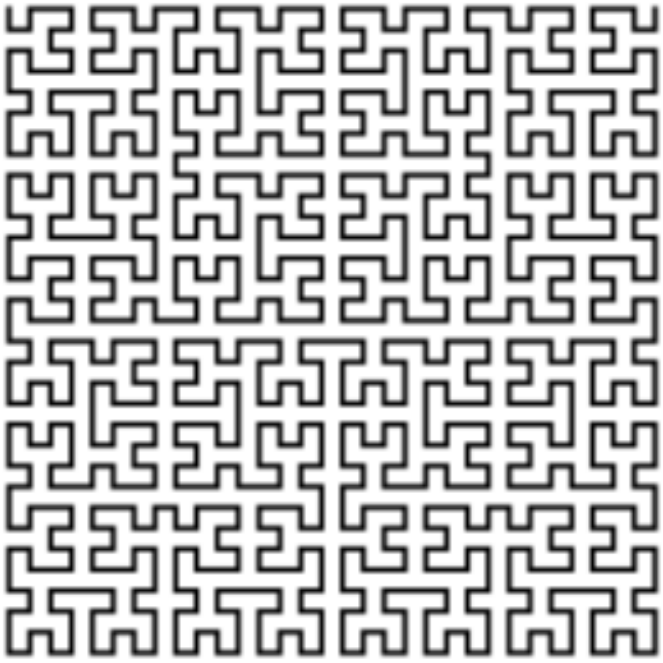

*Рис.2.*1. *Графічне зображення кривої Гільберта.*

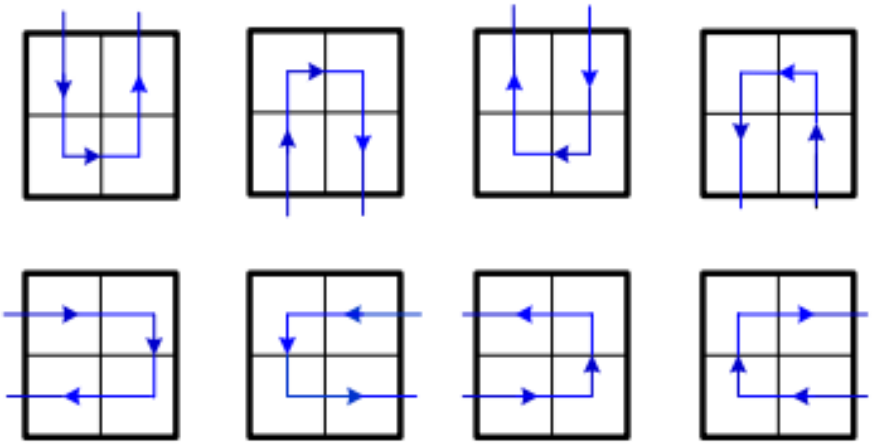

Рис.2.2 – Вісім різних схем скремблювання

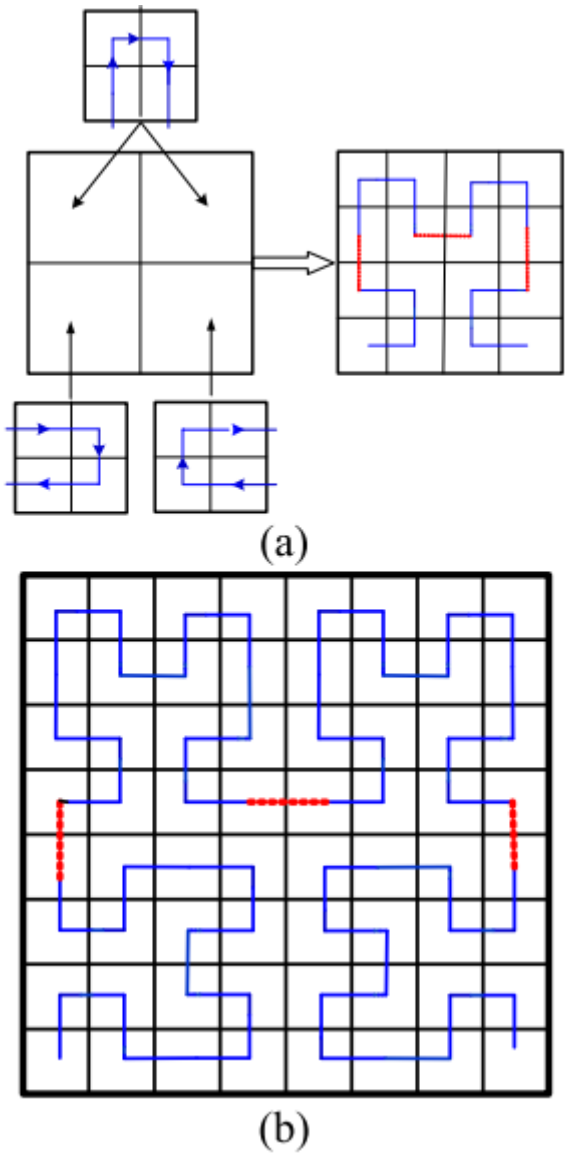

Рис.2.3 Процес генерації двовимірної кривої Гільберта,

(а) друга ітерація кривої Гільберта, (б) двовимірна крива Гільберта.

Від самого спочатку засновник фрактальної геометрії Бенуа Мандельброт визначив фрактал як множину. Згодом Кеннет Фальконер у своїй праці «Фрактальна геометрія: математичні основи та застосування» [19] значно розширив методи застосування фрактальної геометрії. Відтоді вона широко використовується в математиці, фізиці, комп'ютерних науках через характеристичні властивості фракталів.

Зокрема, у фрактальній геометрії часто використовуваними є H–фрактали, множина Кантора, крива Коха та множина Жулія. H–фрактал – це фрактальна структура дерева, що складається з вертикальних відрізків ліній, і кожен відрізок менший за квадратний корінь з 2 помножити на наступний найбільший сусідній

відрізок. Розмірність Хаусдорфа дорівнює 2, і H–фрактал доволі близький до кожної точки в прямокутнику. Його основні застосування включають шифрування інформації, мікрохвильову інженерію та дизайн великомасштабних інтегральних схем.

Кусково–лінійна хаотична карта (КЛХК). Система КЛХК привертає все більше уваги в алгоритмах шифрування через його меншу чутливість до зовнішніх втручань, ніж звичайна логічна карта, її простоті в представленні, ефективність у реалізації, а також хороша динамічна поведінка. Математичний опис одновимірної КЛХК можна показати таким чином:

$$F(s) = \begin{cases} \dfrac{s}{p}, & if\ 0 \le s < p \\ \dfrac{s-p}{0.5-p}, & if\ p \le s < 0.5 \\ 1-s, & if\ 0.5 \le s < 1, \end{cases}$$

де $s \in [0,1]$, коли контрольний параметр $p \in [0, 0.5]$, КХЛК переходить у хаотичний стан. Система КЛХК має рівномірний інваріантний розподіл і дуже хорошу ергодичність, заплутаність та визначеність, отже може забезпечити відмінну випадкову послідовність і згенеровані за допомогою неї послідовності використовується для глобального кодування позиції пікселя, що покращує безпеку всієї криптосистеми. Фазова діаграма системи КЛХК з $p = 0{,}1$ зображено на рисунку нижче.

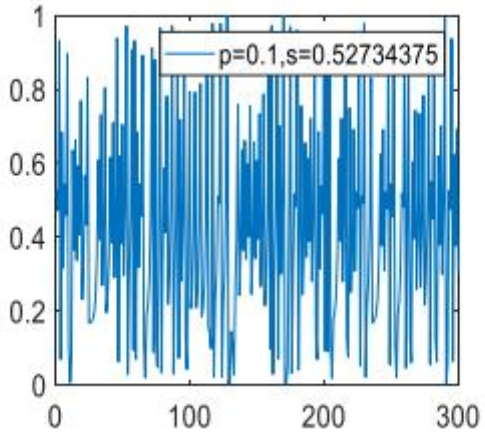

Рис.2.4 – Фазова діаграма системи КЛХК з $p = 0{,}1$

Одновимірна дискретна хаотична система має переваги простої форми та короткого часу генерування хаотичної послідовністі, але її недоліком є те, що простір для ключів занадто малий.

Гіперхаотична система в свою чергу має більш складну динамічну поведінку та має сильніший захист від перешкод і стійкість проти дешифрування. Однак, оскільки гіперхаотична система є більш складною, ніж низько вимірна, збільшення часу для генерування гіперхаотичної послідовності може безпосередньо впливати на вимоги захищеного зв'язку в реальному часі. У процесі шифрування зображення гіперхаотична система використовується для збільшення ключового простору.

Гіперхаотична система Росслера є нелінійною динамічною системою. Він має характеристики непередбачуваного руху траєкторії, чутливість до контрольних параметрів і початкова значення та обмеженість траєкторії руху. Ці характеристики узгоджуються з дослідженнями криптографії тому система широко використовується для шифрування зображень. Рівняння, яке описує гіперхаотичну систему Росслера – $\begin{cases} \dot{x} = -y - z \\ \dot{y} = x + ay + w \\ \dot{z} = b + zx \\ \dot{w} = c\dot{w} - dz, \end{cases}$

де x, y, z і w – змінні стану, a, b, c і d є контрольними параметрами. Коли $a = 0.25,\ b = 3,\ c = 0.05,$ і $d = 0.5$ система перебуває в гіперхаотичному стані.

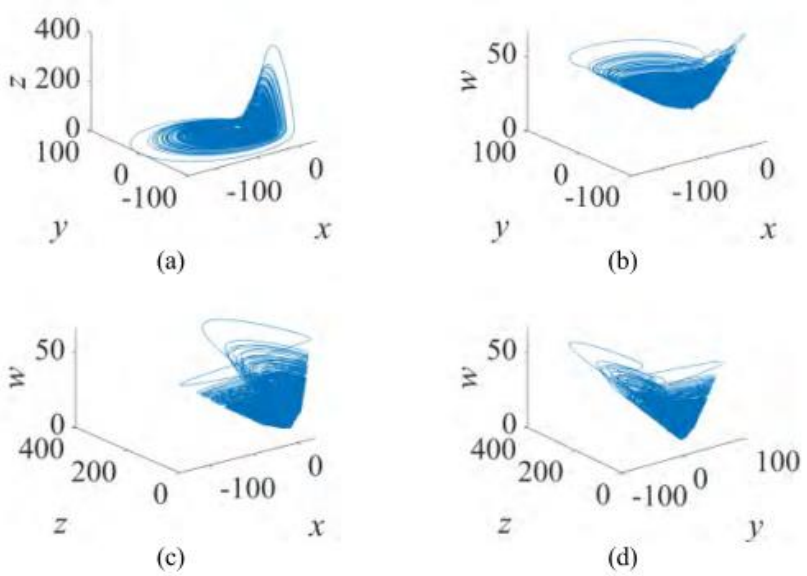

Рис.2.5 – Фазові діаграми різних площин гіперхаотичної системи Росслера

Отже була розглянуто загальна схема алгоритму шифрування, що базується на теорії хаосу, нижче ми розглянемо безпосередні кроки для здійснення шифрування та проведемо аналіз отриманого шифротексту за використанням цього алгоритму. Варто відмітити, що також цілком можливо розглядати шифрування аудіо за допомогою даного алгоритму.

Загалом, в останні пару десятиліть криптографія на основі хаосу користується певною популярністю серед фахівців IT–галузі, оскільки вони забезпечують такі бажані характеристики процесу шифрування, як псевдовипадковість, складність і чутливість до змін параметрів.

2.2. Шифрування мультимедіа за допомогою клітинних автоматів

Розглянемо яким чином здійснюється шифрування за допомогою КА, а також його математичний опис.

У КА другого порядку стан кожної комірки в момент $t + 1$ залежить від сусідства конфігурація стану при t і стан комірки при $t - 1$:

$\varphi^{t+1}(x) = f(\varphi^t(x - r), ..., \varphi^t(x), ..., \varphi^t(x + r), \varphi^{t-1}(x - r), ..., \varphi^{t-1}(x), ...,$

$\varphi^{t-1}(x + r))$, де $\varphi^t(x)$ – стан комірки x у момент t, f – функція локального переходу, а r – радіус околу.

З цією додатковою залежністю та зв'язуванням двох елементарних КА ми можемо будувати двосторонні правила. Два правила елементарних КА R1 і R2 мають бути пов'язані одне з одним таким чином: $R_2 = 2_d - R^1 - 1$, де $d = 2^{2r+1}$. Перше правило визначає перехід стану, коли комірка в момент $t - 1$ перебував у стані «1», а другий, коли комірка була в стані «0». Наприклад вибираючи $R_1 = 30$ і $r = 1$, ми маємо $R_2 = 2^{2^3} - 30 - 1 = 225$, тоді ми можемо представити наше друге правило порядку:

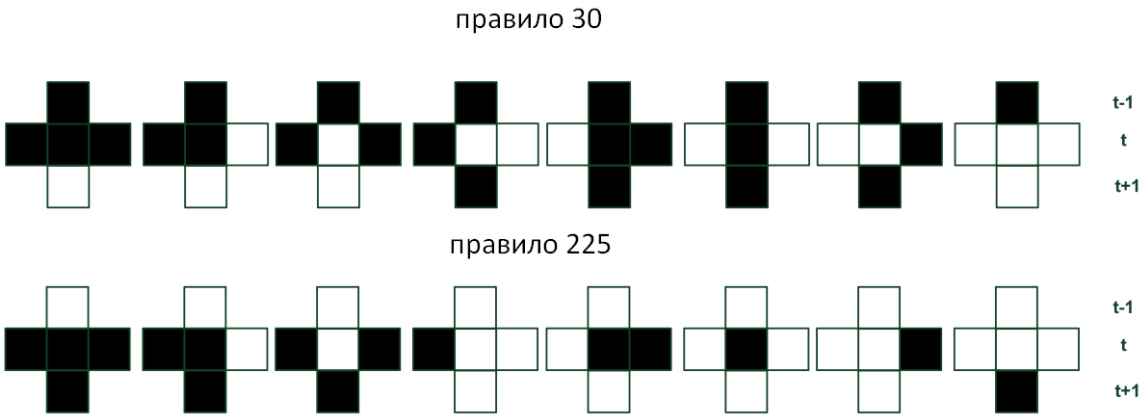

Рис.2.6. Ілюстрація правил заповнення комірки

Далі ми розглянемо, як ці оборотні правила другого порядку можна використовувати в криптосистемі. Після набору перших двох конфігурацій (наприклад, з випадковими даними та відкритим текстом) процес шифрування виконується шляхом розширення КА на попередньо визначену кількість k кроків, як показано на рис.2.7 Коли шифрування завершено, остання конфігурація вважаються залишковими або небажаними даними, для яких можна виконати XOR за допомогою закритого ключа.

В свою чергу передостанньою конфігурацією є наш зашифрований текст. Оскільки кожен крок вимагає двох попередніх конфігурації, ці дві остаточні конфігурації (залишок і зашифрований текст) повинні бути збережено, оскільки вони знадобляться для процесу розшифровки, як дві початкові конфігурації автоматів.

Після цього КА має повторити таку ж кількість кроків, як у процесі шифрування, k кроків, і отримана передостання конфігурація є відновленим початковим текстом.

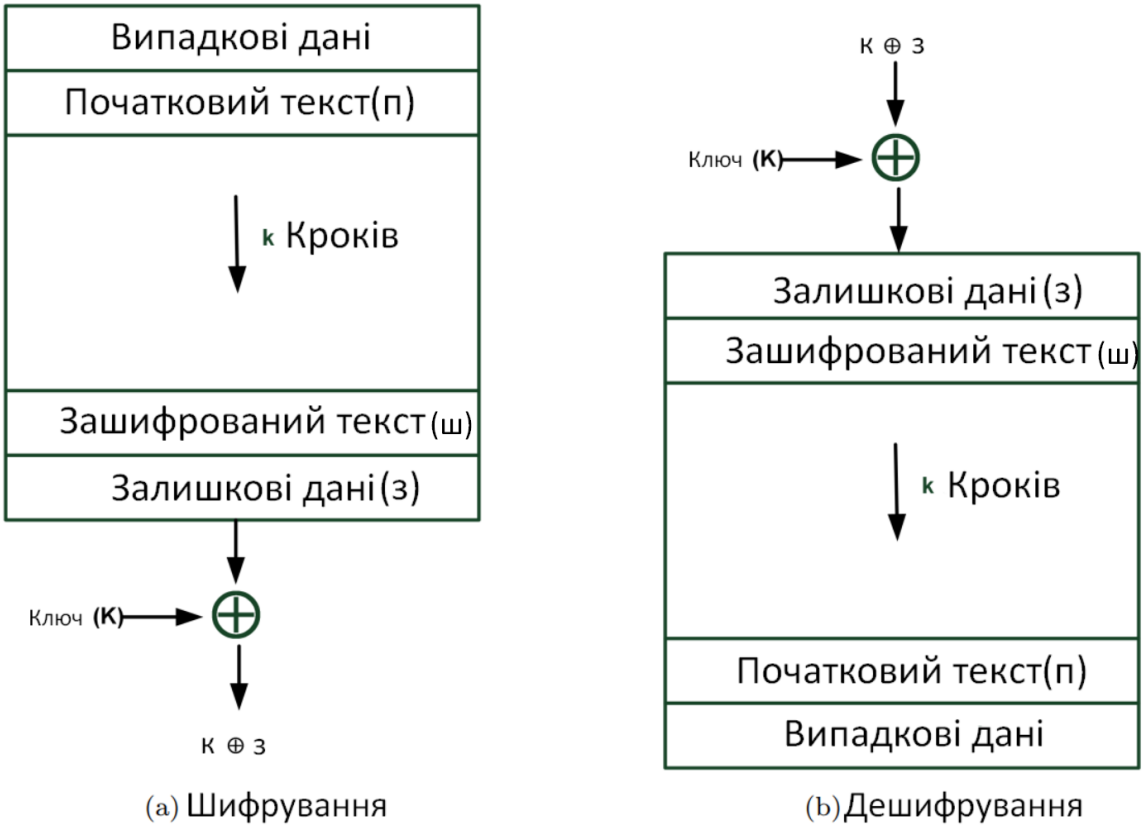

Рис.2.7. Схема шифрування за допомогою КА

Оскільки цілком очевидно, що криптосистема, заснована лише на одному оборотному правилі другого порядку та на кількох ітераціях, не є безпечною, була запропонована більш безпечна криптосистема. У цій системі дві початкові конфігурації СА заповнюються випадковими даними та відкритим текстом, як показано на малюнку. Щоб підвищити безпеку, використовують кілька раундів маніпулювання та перетворення даних.

Відповідно до схеми схемою Серединські–Буврі один її раунд складається з чотирьох одновимірних КА: CA_L, CA_R, CA_C і CA_S (рис.2.3). Усі КА мають радіус 2, крім CA_C з радіусом–3. Кожен блок відкритого тексту має довжину 64 біти, а ключ – 224 біти, що визначає чотири правила RCA, які слід використовувати, і кількість еволюційних кроків. CA_L, CA_R, CA_S складаються з 32 клітин, а CA_C – з 64 клітин. Шифрування кожного відкритого блоку тексту яскладається з попередньо визначеної кількості раундів, де дані зазнають розщеплення, зміщення, рекомбінації та зміни конфігурації в кожному раунді відповідним КА: CA_L і CA_R застосовує правило MCA до лівої та правої частини даних відповідно k_1 разів, CA_C застосовує правило MCA до всіх бітів даних k_2 разів і CA_S генерує

зміну для застосування. Початкові дані генеруються випадковим чином і в режимі CB_C використовується кінцеві дані як початкові для подальшого шифрування блоки. На додаток до збереження згенерованого зашифрованого тексту, остаточних даних і останніх двох також необхідно зберегти конфігурації CA_S. Процес дешифрування виконується в зворотному порядку порядок, включаючи чергування. Кількість еволюційних кроків, необхідних кожному CA для досягнення лавинного ефекту та оптимального ефекту було отримано експериментально оцінка.

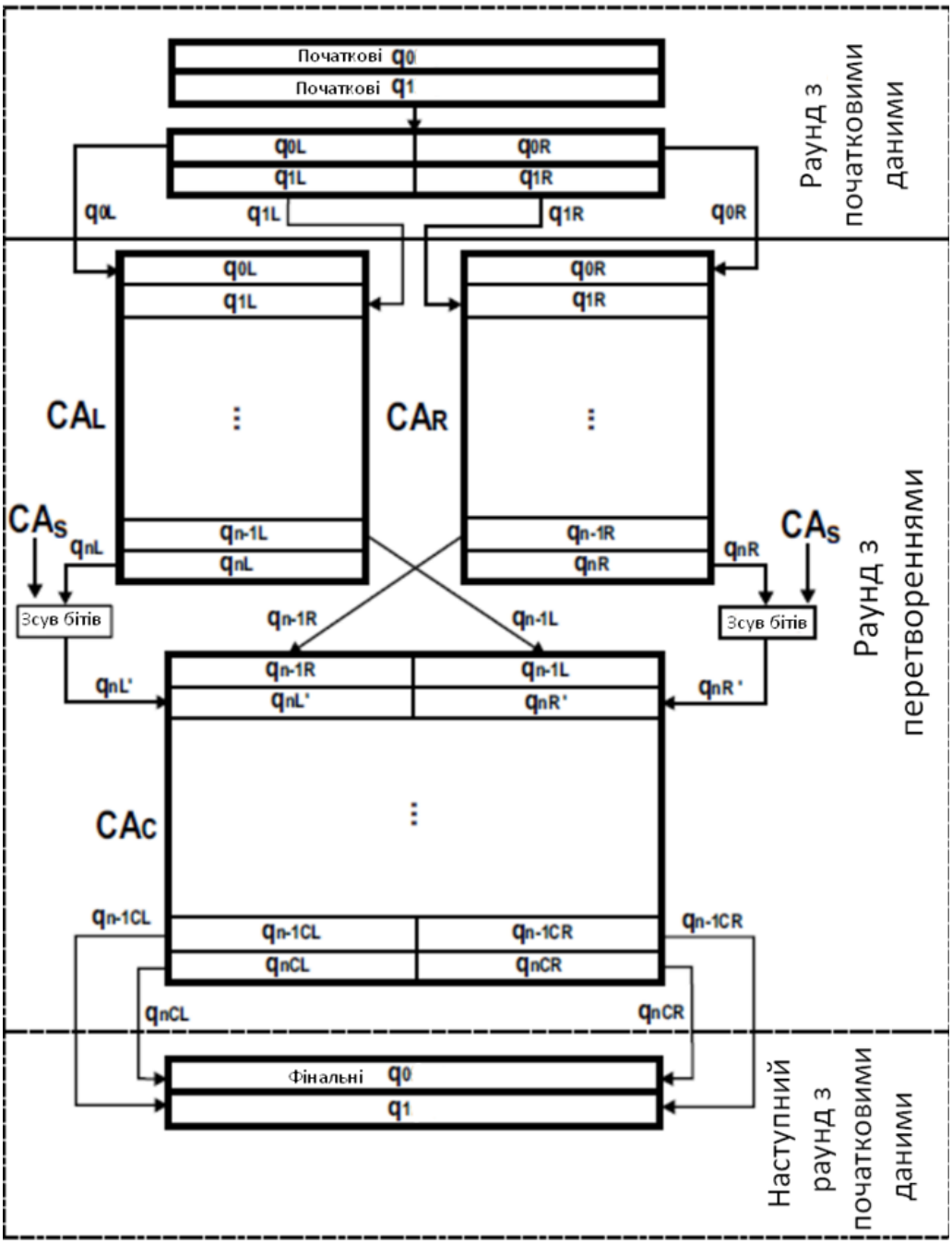

Рис.2.8 Один раунд за блочною схемою Серединські–Буврі[10]

Атаки грубою силою на ключ не можуть бути розглянуті, оскільки розмір ключового простору: 2^{224}. Також обчислювально неможливо виявити початкові конфігурації автомата. Хоча знайти конфігурації CA_C і CA_S (2^{64} і 2^{32} відповідно) обчислювально можливо, зловмиснику все одно доведеться перевірити всі 2^{32} можливі конфігурації. як CA_L, так і CA_R, якщо він використовує підхід грубої сили. Щоб запобігти будь–якій можливості атаки грубою силою, розмір блоку можна збільшити, наприклад, з 64 до 128 біт.

Однією з широко використовуваних систем шифрування зображень є системи хаосу. Деякі особливості системи хаосу включають їх залежність від початкових умов і параметрів керування. Один із способів шифрування зображень на основі хаосу ми розглянули вище. Загалом, стійкість до атак, ці схеми забезпечують 2 речами: плутаниною та дифузією . У фазі плутанини однозначне зображення порушується тим, що кореляція між двома сусідніми пікселями надзвичайно низька, а у фазі дифузії значення пікселя змінюються таким чином, що вплив окремого простого зображення поширюється на зашифроване зображення настільки, наскільки можливо. Крім того, враховуючи аналіз безпеки, такий можна зробити висновок, що послідовність ключа шифрування також має бути пов’язана з пікселями зображення, інакше вона буде чутливою до атак. Крім того, ці методи можуть бути недостатньо чутливими до простого зображення.

Для більшої ефективності обчислень деякі дослідники поєднують такі інструменти, як КА і лінійні регістри зсуву зі зворотним зв'язком. Вони мають достатню складність і допускають паралельне виконання. Нижче ми розглянемо, як використати переваги хаотичної карти та лінійну пам'ять КА (ЛПКА), для шифрування зображення на основі блочного шифрування.

Ми використаємо хаотичну карту для створення псевдовипадкових послідовностей і застосуємо до них КА для розсіювання значення пікселів за допомогою ефективних і паралельних обчислень [6].

Багато з існуючих схем шифрування зображень не може забезпечити можливість паралельної обробки та високу чутливість до змін одночасно. Але, нижче ми намагатимемось забезпечити обидві можливості разом. Крім того, ми

розширюємо механізм автентифікації, а також запобігаємо використанню під час атак грубою силою і може бути налаштований на будь–який бажаний рівень безпеки.

Нехай КА буде визначено як $CA = \{Ç, S, V, Ƒ\}$, де: Ç, клітинний простір, S, набір дискретних станів клітини, де найпростіший стан $S \in \{0, 1\}$, V позначає сусідні комірки, Ƒ – функція переходу, що включає правила для визначення наступного стану клітини. Зробивши припущення, що $a_i^{T+1} = f(N_i^T, 0 \leq i \leq N - 1$.

У наведеному вище рівнянні N_i^T описує стан сусідів комірки i на кроці часу T. Ця формула виражає те, що наступний стан кожної комірки визначається відповідно до функції переходу з введенням її поточного стану та її стану сусідів. Стани всіх комірок на кроці часу T утворюють конфігурацію.

Наразі будемо використовувати реверсивну ЛПКА. У ЛПКА порядку t наступні стани комірок залежать від t попередніх конфігурацій, які визначаються наступним чином:

$$a_i^{T+1} = f_1(N_i^T) + f_2(N_i^{T-1}) + \cdots + f_t(N_i^{T-t+1}) mod\ 2$$

У наведеному вище рівнянні $f_i, i = 1 \ldots t$ є функціями локального переходу в ЛПКА. Якщо $f_t(N_i^{T-t+1}) = a_i^{T-t+1}$ тоді ЛПКА є оборотною через згадану функцію переходу, а її зворотною є інша ЛПКА з наступною локальною функцією переходу:

$$a_i^{T+1} = \sum_{m=0}^{t-2} f_{t-m-1}(N_i^{T-m}) + a_i^{T-t+1}, \qquad 0 \leq i \leq N - 1$$

Щоб забезпечити можливість паралельної обробки, ми спочатку ділимо зображення на кілька блоків. Тоді кожен блок шифруємо за допомогою ЛПКА m–порядку. Алгоритм перестановки виконується на рівні блоку, щоб усунути відношення між послідовними блоками. Цей підхід збільшує швидкість виконання порівняно з перестановкою пікселю або біта. Нарешті виконується другий прохід ЛПКА m–порядку з новими правилами переходу. Щоб збільшити складність методу, правила ЛПКА визначаються псевдовипадковою послідовністю.

Ми генеруємо псевдовипадкову послідовність, використовуючи логістичну хаотичну карту, визначену відповідно до наступної формули:

$$x_i^1 = ax_{i-1}(1 - x_{i-1}), x_{i-1} \in [0, 1]$$

де, a – параметр логічної карти. Вихідна послідовність хаотичної мапи $a \in [3.57, 4]$.

Поки кожен блок зашифровано незалежно, значення його пікселів не впливають на інші зашифровані блоки. Для створення такого ефекту обчислюється хешоване значення зображення з перемежуванням і додається до процесу шифрування. Це значення визначено таким чином, що воно використовується як для автентифікації, так і для шифрування. У третьому розділі буде описано алгоритми шифрування та дешифрування зображення, реалізовані в рамках авторського дослідження.

Також, варто зазначити, що в рамках використання подібних алгоритмів цілком можна говорити про їх екстраполяцію на процес шифрування відео. Оскільки цифрове відео, як відомо, складається з послідовності ортогональних растрових цифрових зображень (кадрів), що відображаються з постійною швидкістю. Тому шляхом певних перетворень відео можна представити набором окремих кадрів. Даний процес є складним та технічно вимогливим. Для його якісного виконання необхідно виконати кілька етапів:

1. Перший етап – це власне отримання відеоданих. Це може включати захоплення відео з камер або читання з файлу, при достатніх обчислювальних потужностях можна розглядати передачу відео в реальному часі.
2. Другим етапом є розбиття на окремі кадри. Розбиття відео потоку на окремі кадри є критичним етапом в процесі перетворення відео в набір окремих кадрів. Цей етап включає в себе використання необхідних методів для розбиття та вилучення окремих кадрів, такими можуть бути:
    a. Метод аналізу руху – виявлення руху для визначення моментів переходу між кадрами.

b. Метод часових інтервалів, що розбиває відео на фіксовані часові інтервали, визначаючи кадри на основі цих інтервалів.
c. Використання мейнфреймів – виділення ключових кадрів, що представляють важливі моменти відео.
d. Методи з використанням алгоритмів машинного навчання для визначення моментів розбиття.

Для уникнення помилкових визначень моментів розбиття можна також використовувати зміни в сцені, враховувати контекст сцени.

3. Третій етап – це обробка кадрів. Кожен окремий кадр за необхідності може бути підданий обробці, такій як фільтрація, збільшення чи зменшення роздільної здатності, або виявлення об’єктів.
4. Четвертий і завершальний етап – це буде збереження окремих кадрів у вигляді окремих файлів для подальшого використання.

Варто відмітити, що всі ці процеси вимагають певних обчислювальних здатностей апаратури, що використовується та певних затрат по часу, що завжди є доступними, саме тому шифрування відеоряду таким чином може залишатись темою для подальших дискусій та досліджень.

Якщо мова йде про інтерактивне відео або потокове мультимедіа, що транслюється в режимі реального часу, то варто попереднього визначати в ньому регіони інтересів (англ. region of interest, ROI) і проводити шифрування лише цих ділянок. Також цілком можливо розглядати шифрування аудіо за допомогою даного алгоритму

Висновок до розділу 2

У 2 розділі було описано особливості підходів до шифрування мультимедійних даних, що спираються на використання окремих математичних структур ТХ та КА, зокрема метод шифрування за допомогою КА із застосуванням блочного шифрування та двосторонніх правил, а також приклад застосування ХС у поєднанні з двома фрактальними функціями. Один з наведених алгоритмів шифрування зображень базувався на властивості заповнення простору кривої Гільберта та властивості нескінченності Н–геометричного фрактала, яка

поєднує в собі псевдовипадковість гіперхаотичної системи. А також блочний алгоритм з використанням КА із лінійною пам’яттю та псевдовипадковою послідовністю, побудованою на основі квадратного різницевого рівняння, що називається логістичною картою.

РОЗДІЛ 3.
АЛГОРИТМИ ШИФРУВАННЯ З ВИКОРИСТАННЯМ КЛІТИННИХ АВТОМАТІВ І ХАОТИЧНИХ СИСТЕМ

3.1. Приклад алгоритму шифрування на основі хаотичної системи

Нижче буде детально розглянутий алгоритм шифрування на основі хаотичної системи. Такий детальний розгляд необхідний для наступного написання коду на мові програмування Python. Дана програма дасть змогу практично застосовувати алгоритм шифрування та експериментувати з ним. Код буде наведений в додатку.

Гіперхаотична система використовується для перестановки та скремблювання пікселів, крива Гільберта використовується для кодування пікселів, а H–фрактальна структура використовується для дифузії. Зрештою, зворотний зв'язок зашифрованого тексту ще більше посилює плутанину та здатність розповсюдження пікселів, таким чином досягаючи шифрування зображення.

Ми використовуємо хеш–функцію для генерації ключа, який є початковим значенням хаотичної системи з метою встановлення асоціації між ключем і відкритим текстовим зображенням.

Вихідне зображення хешується для створення набору 256–бітних хеш–значень, які перетворюється на двійкові значення та використовується як ключ K для початкового значення хаотичної системи. Початкові значення, згенеровані цим методом, мають переваги випадковості та періодичності. K ділиться на байти і може бути розділений на 32 байти, виражені як $k1, k2, \ldots, k32$.

$\mathrm{Q1} = k1 \oplus k2 \oplus \ldots \oplus k8, Q2 = k9 \oplus k10 \oplus \ldots \oplus k16, Q3 = k17 \oplus k18 \oplus \ldots \oplus k24$ і $Q4 = k25 \oplus k26 \oplus \ldots \oplus k32$.

Початкові значення гіперхаотичної системи Росслера і система КЛХК розраховується, як показано нижче

$$\begin{cases} x_0 = Q_1/256 + x'_0 \\ y_0 = Q_2/256 + y'_0 \\ z_0 = Q_3/256 + z'_0 \\ w_0 = Q_4/256 + w'_0 \\ S_0 = Q_4/256 + s'_0, \end{cases}$$

де $x'_0,\ y'_0,\ z'_0,\ w'_0, s'_0$ – початкові значення.

Після хешування слідує етап локального, а згодом і глобального скремблювання пікселів. За допомогою кривої Гільберта, піксельна матриця даного зображення ділиться на чотири підматриці та сканується за допомогою кривої Гільберта. Потім, чотири підматриці діляться на чотири менші підматриці та скануються за допомогою кривої Гільберта. За аналогією, поки кожна підматриця не стане блоком 2 × 2 пікселів. Шлях проходження кривої зберігає значення пікселів матриці зображення A в іншу матриця зображення B. Нова матриця зображення B це піксельна матриця після проходження кривої Гільберта.

Рис.3.1 Зображення після скремблування Гільберта, (a) зображення Олени відкритим текстом, (b) зашифроване зображення, (c) розшифроване зображення Олени.

Для скремблювання використовується система КЛХК для досягнення глобального шифрування. Ми повторюємо систему КЛХК $M \times N$ разів і генеруємо хаотичні послідовності. Сформовані хаотичні послідовності розташовані в порядку зростання, а індекс перестановки послідовності кожного елемента у відсортованій послідовності записується оригінальна хаотична послідовність.

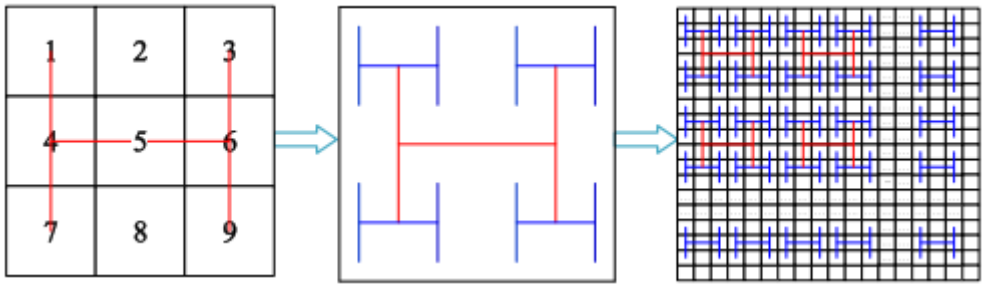

Рис.3.2 – Процес утворення H–фрактала

На цьому етапі використовується H–фрактал, щоб провести локальну дифузію значень пікселів зображення.

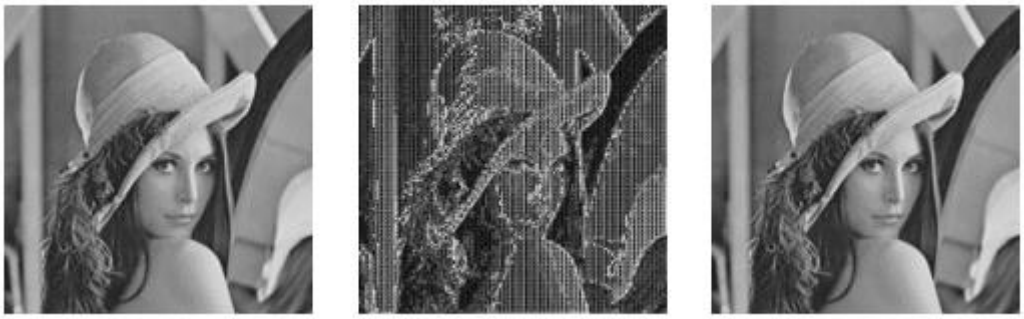

Рис.3.3 – Зображення після H–фрактальних операцій, (a) Зображення Олени в відкритому тексті, (b) Розсіяне зображення, (c) Розшифроване зображення Олени.

Далі операція дифузії шифротексту робить незначні зміни в тексті відкритого тексту, які розповсюджуються на весь шифротекст, тим самим руйнуючи зв'язок між зображенням відкритого тексту та зображенням шифротексту. Це може ефективно захищати від криптографічних атак, таких як атаки на вибраний відкритий текст, та реалізовувати розповсюдження шифротексту. Матриця зображення перетворюється в одновимірну послідовність $S = \{s_1, s_2, s_3, \ldots, s_{M\times N}\}$ довжини $M \times N$ у рядку першого порядку. Потім нехай послідовність дифузія зашифрованого тексту бути SE = $\{se_1, se_2, se_3, \ldots, se_{M\times N}\}$. The дифузія зашифрованого тексту виглядає наступним чином у формулі: se(i) = s(i) $\oplus$ se(i – 1),

де елементи ініціалізації $se(0) = 128$ та $i = 1, 2, \ldots, M \times N$.

Власне алгоритм шифрування цифрового зображення, запропонований вище поєднує сканування кривої Гільберта та H–геометричну фрактальну структуру, яка в основному включає наступні кроки. По–перше, це скремблювання позиції, яке використовує хаотичні послідовності породжені хаотичною картою, для скремблювання та турбувати пікселі зображення. По–друге, сканування кривої Гільберта і H–геометричний фрактал по черзі використовувався для реалізації скремблювання позицій пікселів і дифузії значення пікселів. Нарешті, піксель розсіюється через зворотній зв'язок зашифрованого тексту. Конкретні кроки наведені нижче.

1. Перетворення зображення P у градаціях сірого в двовимірну матрицю зображення P_1 розміром $M \times N$.
2. Отримуємо хеш–значення K матриці зображення P_1і параметри хаотичної ініціалізації x_0, y_0, z_0, w_0 і s_0.
3. Використовуємо одновимірну систему КЛХК для генерації хаотичних послідовностей і проведення глобального скремблювання матриці P_1, і отримання матриці зображення P_2.
4. Повторення створення гіперхаотичної системи Росслера для створення чотирьох хаотичних послідовностей довжиною $M \times N/4$. Потім, взяти цифри від третьої до тринадцятої після коми і провести над ними 256 модульних операцій. $M \times N$ матриця послідовності складається з N елементів у кожному рядку, і матриця послідовності та матриця зображення P_2 піддаються порозрядній операції АБО для отримання матриці розсіяного зображення P_3.
5. Матриця зображення P_4 отримується за допомогою H–фракталу для першої операції дифузії матриці зображення P_3.
6. Матриця зображення P_5 отримується за допомогою сканування Гільбертової кривої в матриці зображення P_4.
7. Формується H–фрактал другого порядку ітерацією кроку 5 повертається за годинниковою стрілкою на 90 градусів, отримуємо деформований H–фрактал. H–фрактал, який деформує зображення матриці P_5 також

використовується для операції дифузії зображення для отримання матриці зображення P_6.

8. Матриця зображення P_7 отримується з фінального раунду операції сканування Гільберта зображення матриці P_6.
9. Виконується операція зворотного зв'язку зашифрованого тексту зі сканованим зображенням кривою Гільберта для отримання матриці P_8, а саме зображення зашифрованого тексту.

Алгоритм дешифрування є зворотним до описаного вище процесу, і алгоритм також застосовний до кольорового зображення шифрування, для якого потрібна лише RGB–декомпозиція значення пікселів зображення.

У додатку подано код, що реалізовує даний алгоритм та виконує шифрування зображень.

Загальний аналіз алгоритму на основі хаотичної системи

Аналіз алгоритмів шифрування в відіграє важливу роль у забезпеченні безпеки та конфіденційності інформації. Цей процес в першу чергу має на меті оцінку безпеки. Аналіз безпеки дозволяє визначити стійкість шифру до різних видів атак. Виявлення вразливостей, тут ми можемо виявляти можливі уразливості алгоритмів, які можуть бути використані для порушення безпеки. Це важливо для подальших вдосконалень та модифікацій. Оптимізація витрат ресурсів та визначення способу використання. Ефективність алгоритму шифру грає важливу роль в практичних застосуваннях, особливо в областях, де ресурси, такі як обчислювальна потужність чи споживана енергія, обмежені. Аналіз допомагає визначити, як правильно використовувати алгоритм шифрування, включаючи налаштування параметрів та ключів, щоб максимально використовувати його потенціал і забезпечити оптимальний рівень безпеки.

Ключовий простір.

Ключі шифрування в цьому алгоритмі включають x0, y0, z0, w0, l0, і 256–бітні хеш–значення. Для початкових значень $x0, y0, z0$ і $w0$ гіперхаотичної системи Росслера та початкового значення $s0$ системи КЛХК, розмір ключового

простору – 10^{40}, а ключового простору – 256–бітового хеш–значення становить 2^{128}, що означає, що загальна сума ключового простору системи шифрування $S =$
$= 10^{40} \times 2^{128} = 3{,}40 \times 10^{78}$. З високою вірогідністю алгоритм буде протистояти атакам грубою силою.

Чутливість ключів до початкових умов

Щоб перевірити чутливість ключів, для гіперхаотичної системи Росслера , змінимо початкові значення $x0, y0, z0$ і $w0$. Рисунок нижче показує відповідне розшифроване зображення.

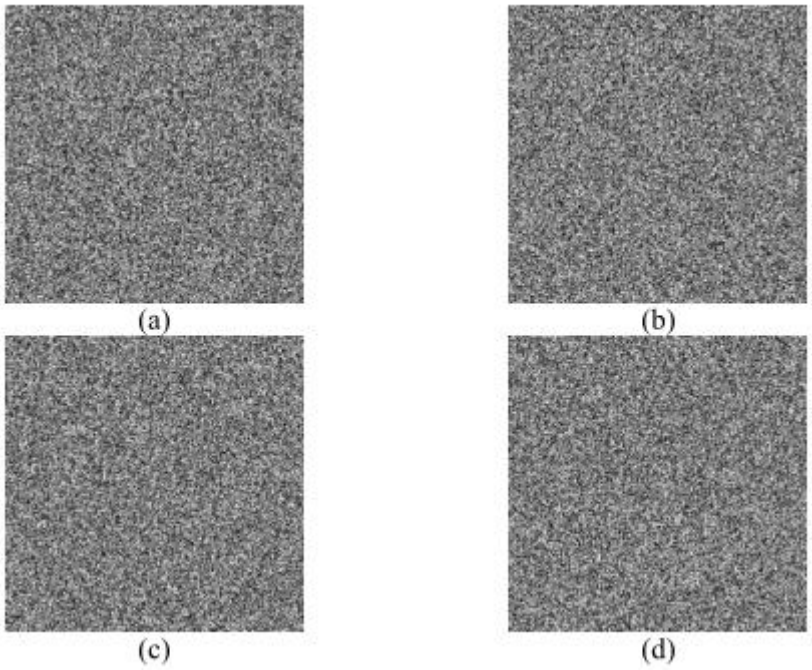

Рис.3.4 Діаграми розшифровки після невеликих змін у ключах.

Оригінальний ключі та модифіковані ключі застосовуються для розшифровки трьох зображення. Можна побачити, що невеликі зміни в ключах не можуть правильно розшифрувати вихідне зображення, тому алгоритм має сильну чутливість до ключів

Аналіз гістограми

До певного рівня статистичні характеристики зображень можуть відображати розподіл відтінків сірого кольору на початкових зображеннях. Те, чи може бути змінений статистичний розподіл початкових зображень, також є важливим показником оцінки в області шифрування зображень.

Рисунок нижче відображає гістограму шифрованого зображення.

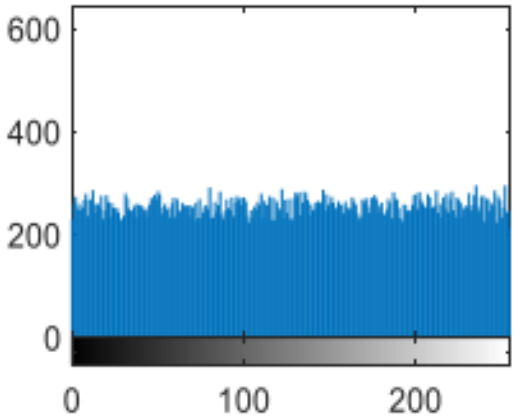

Рис. 3.5 Гістаграма шифрованого зображення

Гістаграма свідчить про те, що алгоритм виявляє високу стійкість до статистичних атак, і атакуючий не може проаналізувати діапазон початкового розподілу відтінків сірого.

Кореляція суміжних пікселів

Дана характеристика є важливим індексом оцінки цифрового аналізу безпеки зображення, що відображає ступінь перемішування розподілу пікселів на зображенні. Чим менша кореляція сусідніх пікселів у зашифрованому зображенні, тим кращим є ефект скремблювання, і навпаки, тим гірший ефект скремблювання.

Кореляція між сусідніми пікселями простого зображення дуже сильна, і таким чином зловмисники можуть легко отримати інформацію відкритого тексту різними засобами. Мета використання шифрування зображення полягає в зменшенні кореляції між пікселями та отримати відповідне зашифроване зображення.

Коефіцієнти кореляції суміжних пікселі між простим зображенням і зашифрованим зображенням показані в таблиці 1, яка показує, що алгоритм має хорошу безпеку.

Таблиця 1 - Коефіцієнти кореляції, середні NPCR, UACI

Зображення	Горизонталь		Вертикаль		Діагональ		Середнє NPCR	Середнє UACI
	Поч.	Шифр.	Поч.	Шифр.	Поч.	Шифр.		
Lena	0.9644	0.0015	0.9332	–0.0014	0.9136	–0.0033	99.564	33.442
Peppers	0.9694	–0.0035	0.9651	0.0050	0.9419	0.0025	99.629	33.380
Cameraman	0.9524	0.0072	0.9194	–0.0113	0.9031	–0.0040	99.645	33.671

Диференціальна атака

Під час цієї атаки зловмисник незначно змінює початкове зображення та порівнює його зашифроване зображення з оригінальним, щоб знайти суттєвий зв'язок. Як правило, два критерії: швидкість зміни кількості пікселів (NPCR) і уніфікована середня змінна інтенсивність (UACI), використовується для визначення того, чи може метод шифрування протистояти диференціальним атакам. Чим ближче характеристика до 100%, тим чутливіша схема шифрування зображення до початкового образу, і тим сильніше його здатність протистояти диференціальним атакам.

В таблиці 1 представлені зведені результати оцінки NPCR та UACI . Як видно з призводить до того, що ця схема має сильний опір диференціальним атакам.

3.2. Приклад алгоритму шифрування з використанням клітинних автоматів

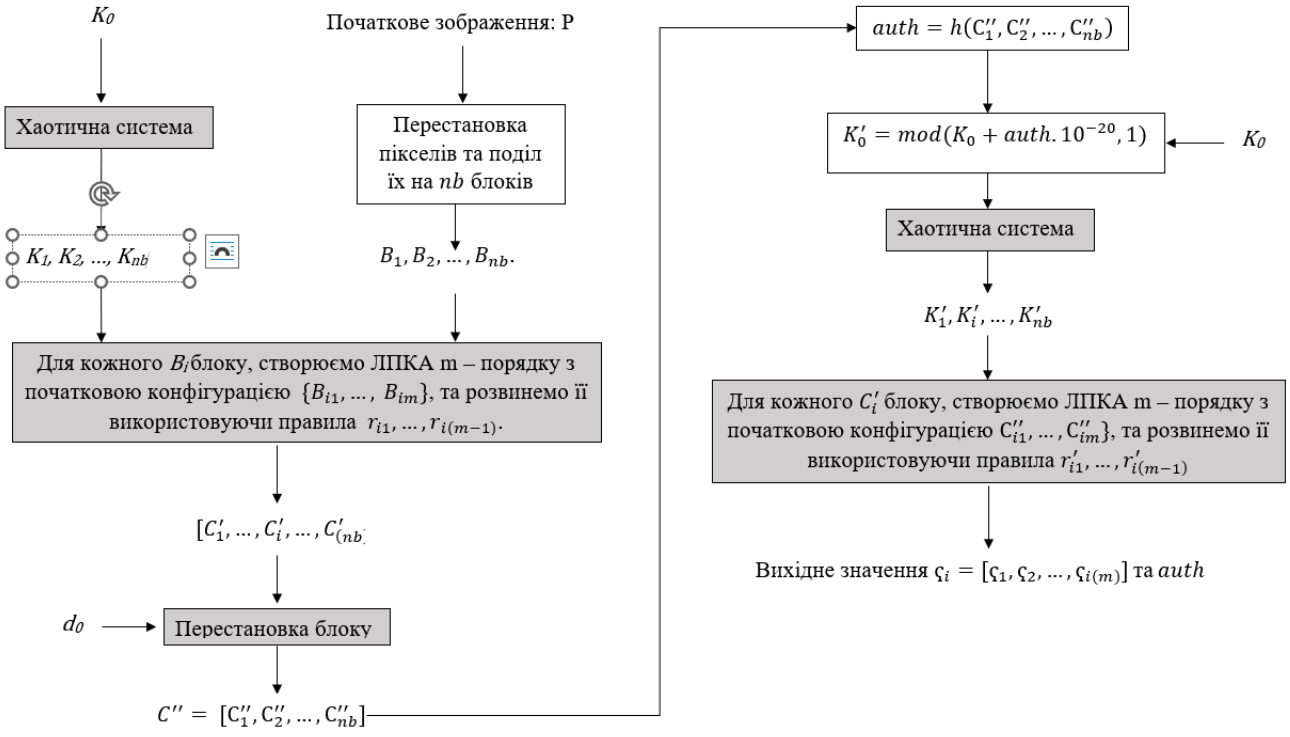

Рис. 3.6 – Схема алгоритму шифрування

Тут припускається, що початкове зображення P є зображенням розміром $M \times N$. Припустимо, що кількість блоків дорівнює nb і розмір кожного блоку $m \times n$. Тоді алгоритм шифрування буде виглядати наступним чином:

1. Розглянемо пікселі зображення P у формі масиву:

$$P = [P_0, P_{nb}, P_{2nb}, \dots, P_1, P_{nb+1}, P_{nb-1}, \dots, P_{r.c}]$$

,де пікселі зображення нумеруються зверху вниз і зліва направо. Потім починаючи з початку масиву, ставимо кожен $m.n$ пікселів в окремі блоки розміром $m \times n$. Завдяки цій простій реорганізації сусідні пікселі, не будуть у блоці. Припустимо, що блоки представлені $B_1, B_2, \dots, B_{nb}$.

2. Розглянемо значення K_0 та d_0, як ключі шифрування. K_0 це початкове значення для генерування ЛПКА правил та d_0 як початкове значення для генерування послідовності перестановки блоку.
3. Повторимо ЛПКА, починаючи з початкових значень K_0 для створення nb чисел [K_1, K_2, …, K_{nb}]. Кожна з цих цифр буде використана для створення ЛПКА правил в блоку.
4. Тоді для кожного блоку B_1 $(i = 1,2, \dots, nb)$, виконаємо наступне:
   4.1 Побудуємо ЛПКА $m-$ порядку з початковою конфігурацією $C_i^0 = = \{ B_i(1), B_i(2), \dots, B_i(m)\}$, де $B_i(j)$ це $j-$ рядок в $i-$ блоці.
   4.2 Створимо набір правил $R_i = \{ r_{i1}, r_{i2}, \dots, r_{i(m-1)}\}$ використовуючи початкове значення K_i таке як: $[r_{i1}, r_{i2}, \dots, r_{i(m-1)}] = f(K_i)$, де f, це функція, що призначає випадкову послідовність бітів правилам ЛПКА.
   4.3 Будемо розвивати ЛПКА прийманні m −разів, починаючи з початкової конфігурації, щоб отримати остаточну конфігурацію $C_i' = = \{C_{i1}', C_{i2}', \dots, C_{i(m)}'\}$.
5. На цьому етапі на блоках $[C_{i1}', C_{i2}', \dots, C_{i(m)}']$ зображення виконується перестановка, використовуючи ключ d_0 отримуємо зображення C''.
6. Обчислюючи хешоване значення $auth$ використовуючи перетворена зображення C'', як $auth = h(C_1'', C_2'', \dots, C_{nb}'')$. $h: Z_{256}^* \to Z_{256}^l$ – хеш – функція,

стійка до зіткнень, де l необов'язковий параметр і залежить від бажаної стійкості авторизації.

7. Створення нового початкового значення K_0' для генерації правил другого виконання ЛПКА:
8. Створення нової послідовності $K_1', \dots, K_{nb}'$ і локальних правил $\{r_{i1}', r_{i2}', \dots, r_{i(m-1)}'\}$, в точності як в кроках 3 та 4
9. Для кожного блоку виконайте ЛПКА принаймні разів із початковою конфігурацією m разів з початковими $\mathrm{C}_i'' = \mathrm{C}_{i1}'', \mathrm{C}_{i2}'', \dots, \mathrm{C}_{im}''$ і локальними правилами $\{r_{i1}', r_{i2}', \dots, r_{i(m)}'$; В кінці фінальна конфігурація $\varsigma_i = = [\varsigma_1, \varsigma_2, \dots, \varsigma_{i(m)}]$ буде отримана.
10. Фінальний зашифрований текст буде отриманий як $CI = \{\varsigma_1, \varsigma_2, \dots, \varsigma_{nb}, auth\}$, тут $auth$ використовується для створення послідовності ключі, а також для виявлення помилок.

Щоб дешифрувати зображення $CI = \{\varsigma_1, \varsigma_2, \dots, \varsigma_{nb}, auth\}$, за допомогою ключа K_0, d_0 кроки алгоритму шифрування потрібно виконати в зворотному порядку. Для правильності зображення шифру ми повинні обчислити значення

$H = h(\mathrm{C}_1'', \mathrm{C}_2'', \dots, \mathrm{C}_{nb}'')$,

після розвитку зворотної ЛПКА в першому раунді. Якщо $H = aut$р, зображення автентифіковано.

Загальний аналіз алгоритму з використанням клітинних автоматів

Ключовий простір

Довжина ключа K_0, d_0 становить принаймні 150 біт, тому ключовий простір становить принаймні 2^{150}. Таким чином цей метод також є стійким до атак грубою силою.

Чутливість ключів до початкових умов

Експериментальні результати показують, що цей метод чутливий до початкового ключа. Наприклад, ми розшифрували з дещо іншими значеннями

ключів і отримали зображення. Порівнюючи відновлені зображення з оригінальними зображеннями, встановленно що їхні пікселі абсолютно різні.

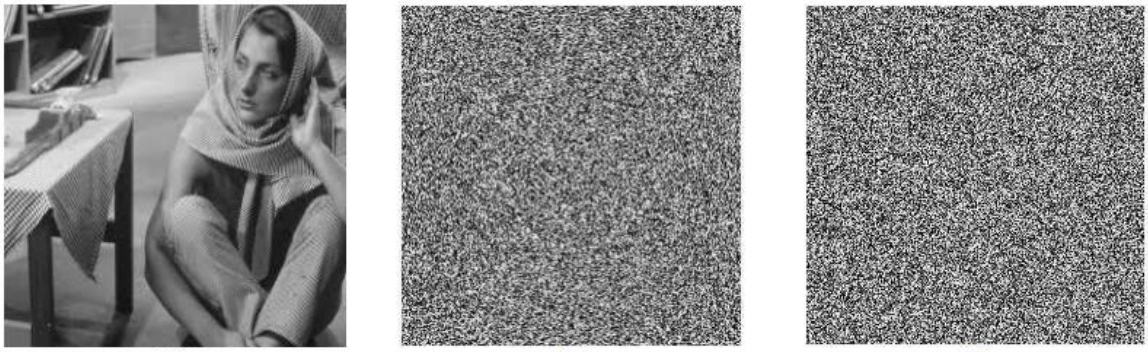

Рис.3.7 (a), (b) Звичайне зображення та відповідне зашифроване зображення; (c) невдало розшифроване

Аналіз гістограми

З гістограми видно, що немає суттєвих точок, за які зловмисник міг би дістати інформацію

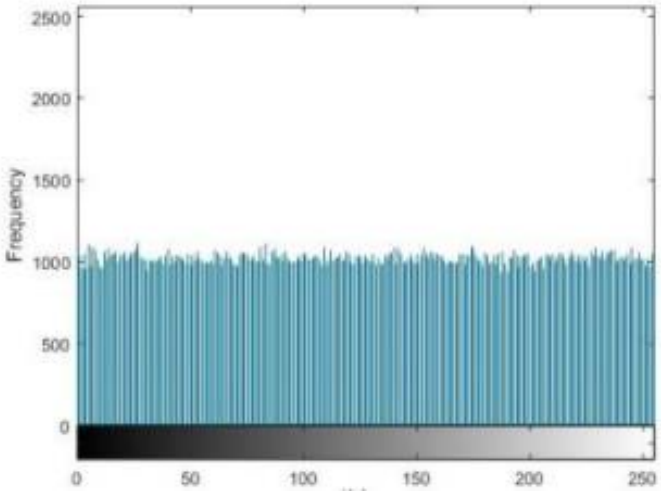

Рис. 3.8 Гістаграма шифрованого зображення

Кореляція суміжних пікселів

Кореляційний аналіз сусідніх пікселів. Пікселі простих зображень часто мають значну залежність, але залежність пікселів шифрованого зображення повинна бути незначною. Коефіцієнти кореляції сусідніх пікселів у деяких початкових та зашифрованих зображеннях зведені в таблиці нижче.

Таблиця 2 – Коефіцієнти кореляції, середні NPCR, UACI

Зображення	Горизонталь		Вертикаль		Діагональ		Середнє NPCR	Середнє UACI
	Поч.	Шифр.	Поч.	Шифр.	Поч.	Шифр.		
Barbara	0.9132	0.0091	0.936	–0.0512	0.9062	0.1278	99.606	33.461
Peppers	0.9655	–0.0027	0.9761	0.0712	0.9428	0.0089	99.594	33.361
Cameraman	0.9597	0.0355	0.9381	–0.0019	0.9325	0.0189	99.581	33.521

Диференціальна атака

Представлений алгоритм надійний проти цієї атаки, оскільки ЛПКА та хеш–функція дають можливість будь–які змінам у своїх вхідних даних впливати на їх кінцевий вихід. Зведення отриманих результатів представлено в таблиці 2.

3.3. Порівняльний аналіз використовуваних алгоритмів

У цьому розділі проводиться аналіз використовуваних алгоритмів шифрування мультимедійних даних, з метою виявлення їхніх переваг та обмежень. Охоплюючи широкий спектр розглянутих методик, розділ ставить перед собою завдання виокремити та проаналізувати ключові аспекти алгоритмів, до прикладу таких, як якість відтворення, операційна швидкість та рівень захисту від несанкціонованого доступу. Для досягнення цієї мети, розділ буде базуватися на результативних дослідженнях та порівняльних аналізах, що були проведені вище.

Здійснюючи об'єктивний огляд кожного алгоритму, розділ покликаний визначити оптимальні стратегії використання в залежності від конкретних вимог та завдань. Перш ніж ми порівняємо алгоритми, давайте порівняємо власне ХС та КА в розрізі криптографії таблиці 3.

Таблиця 3 – Критерії порівняння КА і ХС

Критерій порівняння	КА	ХС
Тип алгоритму	Використовують локальні правила для обробки даних у вигляді клітин або областей	Орієнтовані на динамічні, непередбачувані системи, де дрібна зміна може мати значущий вплив
Стійкість до криптографічних атак	Стійкість залежить від правил обробки та можливості генерації ключів	Оскільки вони чутливі до початкових умов, алгоритми доволі стійкі

Критерій порівняння	КА	ХС
Складність реалізації	Зазвичай відносно прості для реалізації, особливо в контексті обробки зображень	Вимагає глибоких знань математики та фізики для якісної реалізації
Розмір ключа	Залежить відконкретної реалізації, але може бути меншим у порівнянні із хаотичними системами	Великий розмір ключа для забезпечення стійкості
Обчислювальна складність	Зазвичай обчислювально ефективні	Може вимагати значних обчислювальних ресурсів, особливо при використанні складних систем
Використання на практиці	Застосовуються в сучасних криптосистемах та обробці зображень	Знаходить застосування в хаотичних генераторах та системах шифрування

Результат такого порівняння не дасть нам чіткої відповіді, на те, який з підходів кращий. Висновок, який ми можемо зробити полягає в тому, що кожен з підходів має свої переваги та недоліки[32], [13]. Таким чином вибір ХС, або КА буде залежати від конкретних задач та доступних ресурсів.

Далі буде представлена порівняльна таблиця безпосередньо розглянутих алгоритмів. А саме шифрування за допомогою клітинних автоматів та шифрування зображень на основі хаотичних систем в колонках КА та ХС відповідно.

Оцінювання буде проведено, відносно самих алгоритмів шифрування, не враховуючи інші методи, що використовуються на даний момент. В таблиці можна побачити оцінки за шкалою від 1 до 4. Де, 1 погано, 2 – задовільно , 3 – добре, 4 – дуже добре.

Таблиця 4 – Порівняння розглянутих алгоритмів

№	Критерій порівняння	КА	ХС
1	Стійкість до атак грубою силою	3	4
2	Чутливість ключів до початкових умов	4	4
3	Аналіз гістограми	4	3
4	Кореляція суміжних пікселів та диференціальна атака	4	3
5	Обчислювальна складність	3	2
6	Прикладне використання	4	3

7	Гнучкість алгоритму(можливість його модифікації)	2	4
8	Простота реалізації	3	2

Тепер маючи результати ми можемо підсумувати їх та визначити, використання якого з алгоритмів є більш доцільним. Так результуюча оцінка для КА – 27, тоді як для ХС – 25, при максимальній 32. Тепер видно, що представлені алгоритми не мають значної переваги один над одним. Так в першому критерії хоч ХС має більшу оцінку ніж КА, але ні один з алгоритмів не зможе бути зламаний грубою силою в реальних умовах [7]. При аналізі гістограми, ми опирались на візуальні дані, що були отримані, з рисунків 3.5 та 3.8 видно, що КА, мають майже непомітну перевагу, так як графік виглядає гладкішим. Що стосується 4 пункту, то були порівняні результати з таблиць 1 та 2, на яких також можна малу, але перевагу КА. В обчислювальній складності КА також отримали кращу оцінку, оскільки використовує менше математичних моделей та операцій, що також дає перевагу у можливості їх прикладного використання та відповідно у простоті реалізації. Однак алгоритм на основі ХС, має можливість його відносно простої модифікації за допомогою заміни початкових значень, а також можливості повністю замінити хаотичну систему, що використовується.

Також важливо нагадати, що алгоритми оцінювались один відносно одного, що робить вищенаведену оцінку релевантну лише для описаних вище алгоритмів.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було досягнено поставленої мети і виконані всі завдання.

Першим етапом під час написання роботи став аналіз сучасної термінології та концепцій сфери шифрування даних аналіз понять, що є необхідними для розуміння теми. Було описано, як вже доволі старі методи та підходи до шифрування даних, наприклад стенографія, асиметричне та симетричне шифрування, блочне шифрування. Так і перспективні, це такі методи як: квантова криптографія, використання хеш-функцій, застосування КА та ХС, що використовуються, як сучасні підходи в криптографії для підвищення безпеки даних і шифрування, пропонують унікальні перспективи та методи генерації безпечних криптографічних ключів і розробки алгоритмів шифрування.

Останні два привернули змогли привернути нашу увагу та стали темою наступних розділів. Були обрані саме вони, оскільки ХС і КА, які відомі своєю високою складністю і непередбачуваністю, що може зробити криптографічні методи, засновані на них, дуже стійкими до атак інженерного методу та атак грубою силою. А відсутність статистичних закономірностей, є дуже важливою якістю, оскільки такі закономірності можуть використовуватися для криптоаналізу. ХС та КА дають змогу створювати криптосистеми, в яких важко передбачити або визначити розподіл даних, що робить їх ефективними для захисту інформації. Можливість паралельної обробки клітинними автоматами дозволяють розглядати багато поточні обчислення, що може бути корисним для ефективної роботи великих обчислювальних систем.

Дослідивши основи кодування, шифрування, знаючи необхідні поняття та терміни, певні традиційні та новітні підходи та методи в криптографії стало зрозуміло, що особливого розгляду варта саме криптографія за використання саме хаотичних систем та клітинних автоматів.

У 2 розділі було досліджено й описано перспективні підходи до шифрування мультимедійних даних, що спираються на використання математичних структур ТХ та окремих КА. Було детально описано конкретні

алгоритми шифрування за допомогою КА із застосуванням блочного шифрування та двостороніх правил. ХС із фрактальними та іншими математичними функціями, що хоч і роблять алгоритм обчислювально важким, але забезпечують належний рівень захисту.

Також в рамках використання даних алгоритмів, було згадано про можливість шифрування відео. Оскільки цифрове відео, як відомо, містить послідовність ортогональних растрових цифрових зображень (кадрів), що відображаються з постійною швидкістю.

В розділі 3 було описано власне схеми алгоритмів шифрування та необхідних для цього кроків. Наступним кроком став їх аналіз. Спочатку загальний, КА та ХС окремо, під час якого було визначено, що кожний з підходів, як КА так і ХС має свої недоліки та переваги. А згодом і порівняльний аналіз представлених алгоритмів.

В результаті такого аналізу було визначено, що наведений алгоритм за використанням КА є кращим, ніж представлений алгоритм з ХС, хоч і не набагато. Оскільки різниця не значна, можна зробити висновок, що кожен з алгоритмів та підходів вартий уваги, й може знайти собі прикладне використання при правильно підібраних для нього задачах та наявних ресурсів.

В кінці була реалізовано практичну задачу із написання алгоритму шифрування за допомогою ХС на мові Python, що дає змогу для його застосування на практиці.

Дана робота надає детальний аналіз та пропонує можливості використання перспективних методів шифрування, що досі досліджуються та мають на меті покращити стан безпеки персональних даних. Викладений матеріал буде цікавий для усіх науковців, що працюють в сфері безпеки даних, а саме криптографії. Проте особливо корисною праця буде для молодих науковців, що лише починають своє знайомство з криптографією, оскільки вона дає необхідні теоретичні знання в сферах ХС та КА, а також практичну реалізацію одного з алгоритму шифрування, що дає змогу для випробування одного з представлених алгоритмів, його можливої зміни та простору для експериментів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Безрук В. М., Смалько О. А., Коваль О. О., Макаренко М. В. Метод використання динамічних перетворень для стиснення, захисту та приховування відеоінформаційних ресурсів в інфокомунікаційних системах. Сучасна спеціальна техніка. 2023. № 3.
2. Коваль О. Дослідження перспективних підходів до кодування мультимедійних даних. Збірник матеріалів наукової конференції здобувачів вищої освіти фізико-математичного факультету Кам'янець-Подільського національного університету імені Івана Огієнка. 1 листопада 2023 року. Кам'янець-Подільський: Кам'янець=Подільський національний університет імені Івана Огієнка, 2023. С.36. URL: http://elar.kpnu.edu.ua/xmlui/handle/123456789/7648
3. Математичні методи криптології навчальний посібник / А. Кожухівський та ін. Київ, 2021, 244 с.
4. Основи криптографії. навчальний посібник. Чернівці, 2008, 188 с.
5. Швець О. Детермінований хаос. Київ, 2010, 93 с
6. A chaos-based image encryption technique utilizing hilbert curves and h-fractals / X. Zhang et al. IEEE access. 2019. Vol. 7. P. 74734–74746. URL: https://doi.org/10.1109/access
7. AES 256 hardware encryption - safe and secure encryption. ZyberSafe. URL: https://zybersafe.com/aes256hardwareencryption/
8. Al-Musawi W. A., Wali W. A., Ali Al-Ibadi M. A. Field-programmable gate array design of image encryption and decryption using Chua's chaotic masking. International journal of electrical and computer engineering (IJECE). 2022. Vol. 12, no. 3. P. 2414. URL: https://doi.org/10.11591/ijece.v12i3.pp2414-2424
9. A.Mokhtar et al. Gliwice. A new chaos advanced encryption standard (AES) algorithm for data security, 2010.
10. Arabnezhad H., Babak S. An Evaluation and Enhancement of Seredynski-Bouvry CA-based Encryption Scheme, 2021.

11. Carmen P.-L., Ricardo L.-R. Notions of chaotic cryptography: sketch of a chaos based cryptosystem. Applied cryptography and network security. 2012. URL: https://doi.org/10.5772/36419
12. Chaos theory and the logistic map. Geoff Boeing. URL: https://geoffboeing.com/2015/03/chaos-theory-logistic-map/
13. Corona-Bermúdez E., Chimal-Eguía J. C., Téllez-Castillo G. Cryptographic services based on elementary and chaotic cellular automata. Electronics. 2022. Vol. 11, no. 4. P. 613. URL: https://doi.org/10.3390/electronics11040613.
14. Dictionary.com. URL: https://www.dictionary.com/
15. Grzybowski J. M. V., Rafikov M., Balthazar J. M. Synchronization of the unified chaotic system and application in secure communication. Communications in nonlinear science and numerical simulation. 2009. Vol. 14, no. 6. P. 2793–2806. URL: https://doi.org/10.1016/j.cnsns.2008.09.028 (date of access: 24.11.2023).
16. Boguta, K,. Sensitivity To Peturbation in Elementary Cellular Automata, from the Wolfram Demonstrations Project, 2011. URL: http://demonstrations.wolfram.com/SensitivityToPeturbationInElementaryCellularAutomata/
17. Chand S., Aggarwal R., Dubey E. A review of image encryption using chaos based techniques. International journal of science and research, 2015.
18. Eslami Z., Kabirirad S. A block-based image encryption scheme using cellular automata with authentication capability. Third international conference of mathematical sciences (icms 2019), Istanbul, Turkey. 2019. URL: https://doi.org/10.1063/1.5136195
19. Falconer K. Fractal geometry: mathematical foundations and applications. Hoboken, NJ, USA: Wiley, 1990.
20. Legal Definitions Dictionary. URL: https://www.lawinsider.com/dictionary
21. Mankar V. H., Mishra M. Review on chaotic sequences based cryptography and cryptanalysis. International journal of electronics engineering, 2011. P. 189–194.

22. Mao Y., Chen G. Chaos-Based image encryption. Handbook of geometric computing. Berlin/Heidelberg. P. 231–265. URL: https://doi.org/10.1007/3-540-28247-5_8.
23. Mahieu, E,.Ikeda Attractor, from the Wolfram Demonstrations Project, 2011. http://demonstrations.wolfram.com/IkedaAttractor/
24. Ilachinski A. Cellular automata – A discrete universe. Kybernetes. 2003. Vol. 32, no. 4. URL: https://doi.org/10.1108/k.2003.06732dae.007
25. Kaur M., Kumar V. Efficient image encryption method based on improved Lorenz chaotic system. Electronics letters. 2018. Vol. 54, no. 9. P. 562–564. URL: https://doi.org/10.1049/el.2017.4426
26. Kocarev L., Lian S. Chaos-based cryptography: theory, algorithms and applications. Springer, 2011. 408 p.
27. Krapež A. An application of quasigroups in cryptology. Researchgate. 2010
28. Petroc T. Volume of data information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025.
29. Pylypiuk T., Smalko O., Koval O. Research of prospective encoding methods for multimedia content. 2022 IEEE 4th International Conference on Advanced Trends in Information Theory (ATIT). DOI: 10.1109/ATIT58178.2022.10024229
30. Robinson, C. (1995). Dynamical systems. 2nd ed. New York : CRC Press, 1995.
31. Steinmetz R. Multimedia systems. New York : Springer-Verlag, 2004. 466 p.
32. Suneja K., Dua S., Dua M. A review of chaos based image encryption. 2019 3rd international conference on computing methodologies and communication (ICCMC), Erode, India, 27–29 March 2019. 2019. URL: https://doi.org/10.1109/iccmc.2019.8819860
33. Teh J. S., Alawida M., Sii Y. C. Implementation and practical problems of chaos-based cryptography revisited. Journal of information security and applications. 2020. Vol. 50. P. 102421. URL: https://doi.org/10.1016/j.jisa.2019.102421

34. Zolfaghari B., Koshiba T. Chaotic image encryption: state-of-the-art, ecosystem, and future roadmap. Applied system innovation. 2022. Vol. 5, no. 3. P. 57. URL: https://doi.org/10.3390/asi5030057
35. What is group theory and how does it relate to cryptography?. Quora. URL: https://www.quora.com/What-is-group-theory-and-how-does-it-relate-to-cryptography.
36. What is elliptic curve cryptography? Definition & faqs | avi networks. Avi Networks. URL: https://avinetworks.com/glossary/elliptic-curve-cryptography/

ДОДАТОК

Програмний код алгоритму шифрування на основі хаотичної системи Росслера

```
import numpy as np
import hashlib
from PIL import Image
import scipy.ndimage

# Крок 1: Перетворення зображення P у градаціях сірого в двовимірну матрицю
P_1 розміром M × N
def image_to_matrix(image_path):
    img = Image.open(image_path).convert('L')  # конвертація в градації
сірого
    matrix = np.array(img)
    return matrix

# Крок 2: Отримання хеш-значення K матриці зображення P_1 та розбиття його
на 4 блоки
def hash_and_split(matrix):
    hash_value = hashlib.sha256(matrix.tobytes()).hexdigest()
    block_size = 32  # розмір блоку в байтах
    blocks = [hash_value[i:i + block_size] for i in range(0,
len(hash_value), block_size)]
    return blocks

# Крок 3: Отримання параметрів хаотичної ініціалізації для системи Росслера
def chaotic_initialization(q_blocks, x_0, y_0, z_0, w_0, s_0):
    x = (int(q_blocks[0], 16) / 256) + x_0
    y = (int(q_blocks[1], 16) / 256) + y_0
    z = (int(q_blocks[2], 16) / 256) + z_0
    w = (int(q_blocks[3], 16) / 256) + w_0
    s = (int(q_blocks[3], 16) / 256) + s_0

    return x, y, z, w, s

# Крок 4: Створення хаотичних послідовностей і проведення глобального
скремблування матриці P_1
def global_scramble(matrix, x, y, z, w, s):
    m, n = matrix.shape
    chaotic_sequence = np.zeros((m, n // 4))

    for i in range(m):
        x, y, z, w, s = rossler_system(x, y, z, w, s)  # гіперхаотична
система Росслера
        chaotic_sequence[i, :] = extract_digits(x, y, z, w, s)

    # Проведення глобального скрембл.вання
    scrambled_matrix = matrix.copy()
    for i in range(m):
```

```
        scrambled_matrix[i, :] = matrix[i, chaotic_sequence[i, :]]

    return scrambled_matrix

# Крок 5: Використання одновимірної системи КЛХК для генерації хаотичних
послідовностей
# і проведення глобального скрембл.вання  матриці P_1 для отримання матриці
розсіяного зображення P_3
def chaotic_scatter(matrix, x, y, z, w, s):
    m, n = matrix.shape
    chaotic_sequence = np.zeros((m, n // 4))

    for i in range(m):
        x, y, z, w, s = rossler_system(x, y, z, w, s)  # гіперхаотична
система Росслера
        chaotic_sequence[i, :] = extract_digits(x, y, z, w, s)

    # Проведення глобального скрембл.вання  та порозрядна операція АБО
    scattered_matrix = np.bitwise_xor(matrix, matrix[:,
chaotic_sequence.astype(int)])

    return scattered_matrix

# Допоміжна функція для гіперхаотичної системи Росслера
def rossler_system(x, y, z, w, s):
    dt = 0.1
    a, b, c, d, e = 0.2, 0.2, 5.7, 0.02, 5.7
    dx = -y - z
    dy = x + a * y
    dz = b + z * (x - c) + w * np.sin(s)
    dw = d * (e + np.sin(s))
    ds = np.cos(s)
    x += dt * dx
    y += dt * dy
    z += dt * dz
    w += dt * dw
    s += dt * ds
    return x, y, z, w, s

# Допоміжна функція для виділення цифр від третьої до тринадцятої після
коми
def extract_digits(x, y, z, w, s):
    digits = np.zeros(10)
    for i in range(10):
        x, y, z, w, s = rossler_system(x, y, z, w, s)
        digits[i] = int(str(x - int(x))[2:13])  # вибір цифр від 3 до 12
(включно) після коми
    return digits

# Крок 6: Отримання матриці зображення P_4 за допомогою H-фракталу
def h_fractal_diffusion(matrix):
```

```
    h_fractal = np.zeros_like(matrix)
    m, n = matrix.shape

    for i in range(m):
        for j in range(n):
            h_fractal[i, j] = h_function(i, j)

    # Проведення операції дифузії
    diffusion_matrix = matrix + h_fractal

    return diffusion_matrix

# Крок 7: Отримання матриці зображення P_5 за допомогою сканування
Гільбертової кривої
def hilbert_scan(matrix):
    hilbert_curve = scipy.ndimage.morphology.distance_transform_cdt(matrix,
metric='taxicab')
    hilbert_matrix =
matrix.flatten()[np.argsort(hilbert_curve.flatten())].reshape(matrix.shape)

    return hilbert_matrix

# Крок 8: Отримання H-фракталу другого порядку та деформація P_5
def second_order_h_fractal_deformation(matrix):
    deformed_fractal = np.zeros_like(matrix)
    m, n = matrix.shape

    for i in range(m):
        for j in range(n):
            deformed_fractal[i, j] = second_order_h_function(i, j)

    # Деформація матриці P_5
    deformed_matrix = matrix + deformed_fractal

    return deformed_matrix

# Крок 9: Отримання матриці P_7 з фінального раунду операції сканування
Гільберта
def final_hilbert_scan(matrix):
    hilbert_curve = scipy.ndimage.morphology.distance_transform_cdt(matrix,
metric='taxicab')
    hilbert_matrix =
matrix.flatten()[np.argsort(hilbert_curve.flatten())].reshape(matrix.shape)

    return hilbert_matrix

# Допоміжна функція для H-геометричного фракталу другого порядку
def second_order_h_function(i, j):
    return np.sin((i + j) / 10) + np.cos((i - j) / 10)

# Крок 10: Операція зворотного зв’язку зі сканованим зображенням Гільберта
```

```
def inverse_hilbert_feedback(encrypted_text, hilbert_matrix):
    m, n = hilbert_matrix.shape
    reshaped_encrypted_text = encrypted_text.flatten()

    # Сортування за зворотнім порядком сканування Гільберта
    sorted_indices = np.argsort(hilbert_matrix.flatten())[::-1]

    # Відновлення матриці P_8
    decrypted_matrix = np.zeros((m, n))
    decrypted_matrix.flat[sorted_indices] = reshaped_encrypted_text

    return decrypted_matrix

# Приклад використання
image_path = "зображення.jpg"
image_matrix = image_to_matrix(image_path)
hashed_blocks = hash_and_split(image_matrix)
x_0, y_0, z_0, w_0, s_0 = 0.1, 0.2, 0.3, 0.4, 0.5
x, y, z, w, s = chaotic_initialization(hashed_blocks, x_0, y_0, z_0, w_0,
s_0)
scrambled_image = global_scramble(image_matrix, x, y, z, w, s)
scattered_image = chaotic_scatter(image_matrix, x, y, z, w, s)
diffused_image = h_fractal_diffusion(scattered_image)
hilbert_image = hilbert_scan(diffused_image)
deformed_image = second_order_h_fractal_deformation(hilbert_image)
final_hilbert_image = final_hilbert_scan(deformed_image)

# Операція зворотного зв’язку зі сканованим зображенням Гільберта
(припустимо, encrypted_text - це зашифрована матриця P_8)
decrypted_matrix = inverse_hilbert_feedback(encrypted_text,
final_hilbert_image)
```

Результат роботи алгоритму

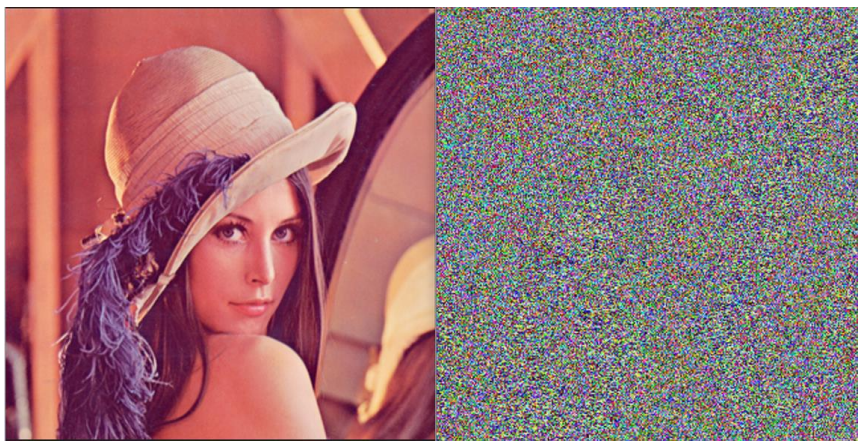

Рис. Початкове зображення(ліворуч), та зашифроване (праворуч)