Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод нанесення цифрових водяних знаків з використанням хаотичних карт

(тема)

Виконав:
здобувач 2 року навчання,
групи СПм-23-5
Антон ШАНДИБА
(власне ім'я, прізвище)

Спеціальність
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: проф. Ігор РУБАН
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Шандибі Антону Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Метод нанесення цифрових водяних знаків з використанням хаотичних карт

затверджена наказом по університету від "21" квітня 2025 р. № 296 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи

Нанесення цифрових водяних знаків із застосуванням хаотичних карт

Програмна реалізація мовою Python з використанням бібліотек OpenCV та NumPy.

4. Перелік питань, що потрібно опрацювати у роботі

Вимоги до цифрових водяних знаків

Методи нанесення цифрових водяних знаків

Властивості хаотичних карток

Використання хаотичних карт у цифровому водяному маркуванні

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Слайд-презентація – 14 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Теоретичні основи цифрових водняних знаків	22.04.25	
2	Хаотичні карти як основа для захисту	23.04.25-28.04.25	
	інформації		
3	Програмна реалізіція алгоритму	29.04.25-10.05.25	
4	Проведення експериментальних досліджень	11.05.25-20.05.25	
5	Оформлення кваліфікаційної роботи	21.05.25-05.06.25	
6	Подання кваліфікаційної роботи керівникові та	06.06.25-09.06.25	
	її попередній захист		
7	Подання кваліфікаційної роботи на рецензу-	10.06.25-12.06.25	
	вання		

Дата видачі завдання " 21 " квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Ігор РУБАН
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи : 68 с., 13 рис., 2 табл., 1 дод., 13 джерел.

ХАОТИЧНІ КАРТИ, КРИПТОГРАФІЯ, ГЕНЕРАЦІЯ КЛЮЧІВ, ЦИФРОВИЙ ВОДЯНИЙ ЗНАК, СТЕНОГРАФІЯ.

Об’єкт дослідження – системи інформаційної безпеки, що використовують хаотичні математичні моделі.

Предмет дослідження – процеси та алгоритми використання хаотичних карт для генерації криптографічних ключів і вбудовування цифрових водяних знаків.

Мета роботи – проаналізувати та обґрунтувати ефективність застосування хаотичних карт у задачах криптографії та цифрового водяного маркування з метою підвищення рівня захисту інформації.

ABSTRACT

Master’s thesis: 68 pages, 13 figures., 2 tables, 1 appendices, 13 sources.

CHAOTIC MAPS, CRYPTOGRAPHY, KEY GENERATION, DIGITAL WATERMARK, STENOGRAPHY.

The object of the study is information security systems that use chaotic mathematical models.

The subject of the study is the processes and algorithms of using chaotic maps for generating cryptographic keys and embedding digital watermarks.

The purpose of the work is to analyze and justify the effectiveness of using chaotic maps in cryptography and digital watermarking tasks in order to increase the level of information protection.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ЦВЗ – цифрові водяні знаки

AES – розширений стандарт шифрування (англ. Advanced Encryption Standard)

DCT – дискретне косинусне перетворення(англ. Discrete Cosine Transform)

DFT – дискретне перетворення Фур'є (англ. Discrete Fourier Transform)

DWT – дискретне вейвлет-перетворення (англ. Discrete Wavelet Transform)

LSB – найменш значущий біт (англ. Least Significant Bit)

RSA – криптографічний алгоритм з відкритим ключем (англ. Rivest–Shamir–Adleman)

TLS – безпека транспортного рівня (англ. Transport layer security)

ВСТУП

У сучасному цифровому середовищі питання охорони авторських прав та безпеки інформації стають надзвичайно важливими. Одним з дієвих способів захисту мультимедійного контенту є технологія цифрових водяних знаків, яка дає змогу ідентифікувати та відстежувати незаконне використання цифрових файлів. Водночас, традиційні способи нанесення водяних знаків мають певні недоліки, що стимулює пошук нових рішень. Об'єктом дослідження є технологія цифрових водяних знаків. Предметом дослідження є способи нанесення цифрових водяних знаків з використанням хаотичних карт. У цій роботі розглянуто теоретичні засади цифрових водяних знаків, проведено аналіз існуючих методів їх нанесення, а також детально вивчено можливість застосування хаотичних карт у цій сфері. Запропоновано методику використання хаотичних карт для генерування ключів і вбудовування водяних знаків. Проаналізовано сильні та слабкі сторони цього підходу, а також окреслено перспективи майбутніх досліджень у цій галузі.

Цифрові водяні знаки (ЦВЗ) – це приховані дані, які впроваджуються в цифрові медіа (зображення, відео, аудіо, тексти) для ідентифікації власника та запобігання несанкціонованому копіюванню чи редагуванню. Вони зазвичай невидимі або малопомітні для звичайного користувача, але можуть бути виявлені спеціалізованими засобами. Основна функція ЦВЗ – захист авторських прав та забезпечення безпеки інформації в цифровому просторі.

З розвитком інформаційних технологій та збільшенням обсягів цифрового контенту, постає потреба в нових способах захисту. Звичайні методи захисту, як-от паролі чи шифрування, можуть бути легко зламані або обійдені. Натомість, цифрові водяні знаки є більш надійним методом, оскільки вони не лише захищають вміст, але й дозволяють відстежувати його походження, підтверджувати автентичність, а також виявляти джерела несанкціонованого розповсюдження.

Сучасний цифровий світ вимагає ефективних інструментів захисту авторських прав і боротьби з піратством. З ростом популярності інтернет-платформ і можливостей швидкого розповсюдження цифрового контенту виникає серйозна загроза незаконного використання та копіювання файлів. Нанесення цифрових водяних знаків – один із найефективніших методів забезпечення захисту файлів від таких загроз. Водночас, це дозволяє зберігати цілісність та автентичність даних, адже водяний знак може бути невід'ємною частиною файлу, яку неможливо безпечно видалити або змінити без виявлення.

Хаотичні карти – це математичні структури, що базуються на принципах хаотичної динаміки, які використовуються в криптографії для генерування складних, непередбачуваних послідовностей, які можна використовувати для захисту даних. Хаос, як явище, характеризується високою чутливістю до початкових умов, що дозволяє створювати складні та стійкі до атак системи. У криптографії хаотичні карти застосовуються для створення випадкових чисел, генерування ключів та інших задач, де потрібна висока ступінь непередбачуваності.

Властивості хаотичних карт, як-от детермінованість, чутливість до умов і здатність генерувати складні структури, роблять їх ідеальними для використання в сучасних системах безпеки. Ці карти можуть служити основою для створення водяних знаків, оскільки дозволяють вбудовувати захищені дані таким чином, що їх неможливо легко виявити чи видалити без порушення цілісності файлу.

Використання хаотичних карт для нанесення цифрових водяних знаків відкриває нові можливості для підвищення стійкості та надійності захисту. Завдяки їхнім властивостям можна створювати водяні знаки, які є надзвичайно стійкими до атак і важко виявляються чи змінюються без пошкодження самого файлу. Хаотичні карти можуть бути використані для генерації ключів, які застосовуються для вбудовування водяних знаків у цифрові файли.

Завдяки таким властивостям, як висока непередбачуваність та складність, хаотичні карти можуть забезпечити додатковий рівень безпеки, захищаючи водяні знаки від атак, спрямованих на їхнє виявлення або видалення. Цей підхід дозволяє значно підвищити ефективність захисту авторських прав та безпеки цифрових файлів, особливо в умовах постійного розвитку технологій кіберзагроз.

1 ТЕОРЕТИЧНІ ОСНОВИ ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ

1.1 Поняття цифрових водяних знаків

Цифрові водяні знаки – це процес непомітного вбудовування спеціальної інформації, відомої як цифровий водяний знак, у різні типи цифрового контенту, такі як зображення, аудіо, відео чи текст.

Вбудований водяний знак виконує різноманітні функції, серед яких:

- захист авторських прав;
- автентифікація контенту;
- відстеження розповсюдження матеріалів;
- контроль за цілісністю даних.

Цифровий водяний знак має відповідати кільком ключовим вимогам:

- непомітність. Внесення водяного знаку не повинно помітно змінювати якість контенту для звичайного користувача. Для його виявлення зазвичай потрібні спеціальні інструменти;

- стійкість. Водяний знак має зберігатися в контенті навіть після типових операцій обробки, таких як стиснення, обрізка, фільтрація, додавання шуму чи геометричні трансформації;

- ємність. Водяний знак має можливість містити певний обсяг інформації – від одного біта до більш складних даних, наприклад, серійного номера або веб-посилання;

- безпека. Вбудований знак має бути захищеним від несанкціонованого видалення або спотворення без наявності спеціальних знань чи ключів;

- однозначність. Процедура виявлення водяного знаку повинна точно ідентифікувати наявність знаку і, за необхідності, відновлювати його вміст.

Ефективна система цифрового водяного знакування включає кілька основних компонентів:

- алгоритм вбудовування. Визначає спосіб інтеграції водяного знаку у цифровий контент. Зазвичай використовує секретний ключ для підвищення безпеки;

- алгоритм виявлення. Реалізує процес пошуку і витягу водяного знаку з контенту. Залежно від методу, виявлення може потребувати або не потребувати оригінального (немаркованого) контенту;

- ключ. Секретна інформація, що використовується на етапах вбудовування і виявлення водяного знаку. Наявність ключа ускладнює спроби стороннього видалення або підробки знаку.

Цифрові водяні знаки можна класифікувати за кількома критеріями:

а) за видимістю:

1) невидимі. Водяний знак непомітний для користувача;

2) видимі. Спеціально розміщені поверх контенту (наприклад, логотип компанії);

б) за стійкістю до обробки:

1) стійкі. Зберігаються після застосування атак чи обробки;

2) крихкі. Руйнуються при найменших змінах контенту, що дозволяє контролювати цілісність;

3) напівкрихкі. Витримують певні допустимі зміни, але руйнуються при суттєвій модифікації;

в) за залежністю від оригіналу під час виявлення:

1) сліпі. Для виявлення не потрібен оригінальний контент;

2) напівсліпі. Потребують додаткової допоміжної інформації:

3) несліпі. Вимагають наявності оригінального контенту для зіставлення.

г)за доменом вбудовування:

1) просторові (часові). Знак вбудовується безпосередньо у значення пікселів або аудіосемплів;

2) частотні. Контент перетворюється у частотну область (наприклад, через дискретне косинусне перетворення (DCT) чи дискретне

вейвлет-перетворення (DWT)), і водяний знак вбудовується у відповідні коефіцієнти;

д) за методом вбудовування: використовуються різноманітні математичні, статистичні та хаотичні підходи для розробки стійких і непомітних алгоритмів.

Розуміння теоретичних основ цифрового водяного знакування є критично важливим для побудови ефективних методів захисту цифрового контенту. Вибір методів та їх параметрів повинен базуватися на конкретних вимогах до стійкості, прозорості та ємності водяного знаку.

1.2 Вимоги до цифрових водяних знаків

Цифрові водяні знаки повинні відповідати трьом основним вимогам: стійкості, прозорості та ємності. Між цими характеристиками існує певний компроміс, тому розробникам доводиться шукати баланс залежно від конкретного завдання.

Стійкість означає здатність водяного знака зберігатися у контенті навіть після різних обробок і атак. Водяний знак має залишатися виявляним, навіть якщо файл зазнає стиснення, фільтрації, додавання шуму або геометричних перетворень. Це критично важливо для захисту авторських прав і відстеження поширення цифрової інформації. Однак досягнення високої стійкості часто вимагає внесення більш помітних змін у контент, що може зменшити його якість для кінцевого користувача.

Прозорість передбачає, що водяний знак не має негативно впливати на сприйняття контенту – ні візуально, ні на слух. Хороший водяний знак не створює артефактів, які помітні неозброєним оком або вухом. Щоб оцінити рівень прозорості, часто залучають експертів для проведення суб'єктивного тестування, хоча існують і об'єктивні метрики, наприклад, показник PSNR для зображень і відео. Підвищення прозорості, втім, може знизити як стійкість, так і обмежити кількість інформації, яку можна вбудувати.

Ємність визначає обсяг даних, які можна заховати в цифровому контенті за допомогою водяного знака. Чим більшу кількість інформації потрібно передати – наприклад, серійний номер, ідентифікатор користувача або навіть цілий текстовий блок – тим вища повинна бути ємність системи. Однак спроби вбудувати більше інформації можуть або зробити водяний знак видимішим, або послабити його стійкість до атак і обробки.

Стійкість, прозорість і ємність перебувають у складному взаємозалежному балансі. Зміцнення однієї характеристики часто вимагає компромісів щодо інших. Наприклад, збільшення стійкості може зробити зміни у контенті більш помітними, а підвищення ємності – знизити стійкість водяного знака. Тому перед розробкою системи цифрового водяного знакування важливо визначити пріоритети: що важливіше – захист, непомітність чи обсяг переданої інформації – і відповідно налаштувати параметри.

1.3 Методи нанесення цифрових водяних знаків

Цифрові водяні знаки можуть вбудовуватись у контент двома основними способами: у просторовій (часовій) або частотній області. Кожен підхід має свої особливості, переваги та недоліки, які визначають його придатність для певних застосувань.

У цьому підході водяний знак вноситься безпосередньо у вихідні дані: значення пікселів зображення, аудіосемпли або відеокадри. Для сигналів, таких як аудіо або відео, аналоги просторової області – це часова область.

Одним із найпростіших способів є зміна найменш значущих бітів (LSB). Інформація водяного знака вбудовується у молодші біти пікселів або семплів, що забезпечує високу прозорість, але при цьому створює високу вразливість до атак і змін контенту. Інші методи в цій області базуються на незначній модифікації піксельних значень за певним шаблоном або на зміні статистичних характеристик зображення чи звуку, наприклад середнього

значення яскравості. Існують також методи накладання слабовиражених шаблонів, які зберігаються у контенті та виявляються спеціальними алгоритмами.

Основними перевагами методів просторової області є їхня простота реалізації та можливість досягнення високої ємності вбудовування. Проте недоліки також значні: низька стійкість до стиснення, фільтрації, додавання шуму та навмисних атак, що може призвести до швидкого руйнування водяного знака.

На відміну від просторового підходу, методи частотної області передбачають попереднє перетворення контенту за допомогою математичних операцій, таких як дискретне косинусне перетворення (DCT), дискретне перетворення Фур'є (DFT) або дискретне вейвлет-перетворення (DWT). Після перетворення водяний знак вбудовується шляхом модифікації обраних коефіцієнтів, після чого контент перетворюється назад у вихідну область.

Зазвичай для вбудовування змінюються коефіцієнти середньої частоти, що дозволяє зберігати баланс між стійкістю і непомітністю змін. Часто використовується також псевдовипадкове розташування водяного знака, що ускладнює його виявлення та видалення. Ще один поширений підхід – квантування коефіцієнтів таким чином, щоб вони несли інформацію водяного знака.

Методи частотної області демонструють вищу стійкість до типових атак і обробки, зокрема до стиснення з втратами, і можуть забезпечувати кращу прозорість, якщо вибрано оптимальні частотні діапазони. Разом із тим вони вимагають значно складнішої реалізації та більших обчислювальних ресурсів. У деяких випадках можливість вбудувати велику кількість інформації обмежується вимогами до стійкості та якості контенту.

Вибір між методами просторової та частотної областей залежить від цілей застосування. Просторові методи підходять там, де потрібна простота та висока ємність, а стійкість не є критично важливою. Частотні методи

доцільні тоді, коли головним пріоритетом є збереження водяного знака після обробки чи атак.

Сучасні дослідження прагнуть поєднати переваги обох підходів, створюючи гібридні або адаптивні системи, які враховують особливості контенту та можливі загрози.

Вбудовування цифрового водяного знаку у файл зазвичай складається з кількох ключових етапів (рисунок 1.1):

а) генерування водяного знаку. На цьому етапі створюється унікальний ідентифікатор або код, який надалі буде вбудований у цифровий контент. Водяний знак може бути сформований у вигляді простого бітового шаблону або складного криптографічного коду, залежно від вимог до безпеки та стійкості;

б) вбудовування водяного знаку. Згенерований водяний знак інтегрується в окремі частини файлу без помітної зміни його зовнішнього вигляду або якості. Це може відбуватись шляхом:

1) зміни значень окремих пікселів у зображеннях;

2) модифікації частотних складових у звукових або відеофайлах;

3) непомітної зміни структури текстових документів. Основна мета цього етапу – забезпечити максимальну непомітність водяного знаку для користувача при збереженні його стійкості до можливих атак чи обробки;

в) витягування водяного знаку. На цьому етапі здійснюється зчитування вбудованого ідентифікатора для перевірки автентичності контенту або підтвердження авторських прав. Процес витягування зазвичай базується на спеціальних алгоритмах і може вимагати наявності секретного ключа для правильного розпізнавання та декодування інформації.

Цифрові водяні знаки поділяються на дві основні категорії:

- видимі водяні знаки – наприклад, логотипи або надписи, накладені на зображення чи відео. Вони легко помітні неозброєним оком і переважно виконують функцію відкритого позначення авторства;

- приховані водяні знаки – вбудовуються таким чином, що їх

неможливо побачити без спеціального аналізу або обробки. Вони використовуються для прихованого захисту авторських прав або аутентифікації даних.

Методи вбудовування варіюються від простих операцій, таких як зміна найменш значущих бітів у зображеннях, до складних криптографічних підходів, які підвищують захищеність водяних знаків від спроб видалення, підробки або інших атак.

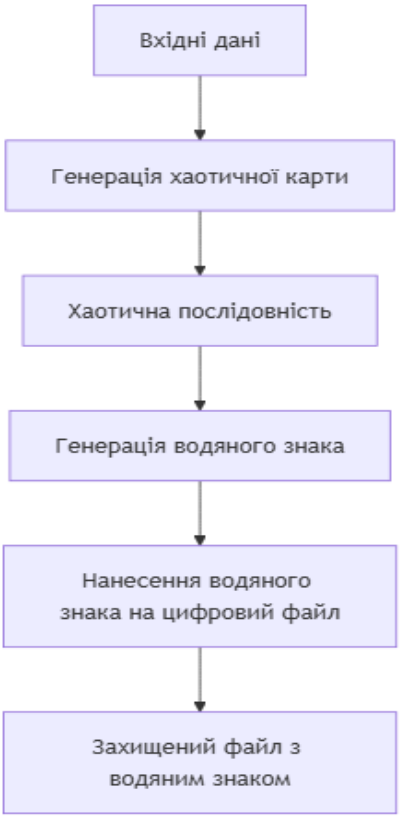

Рисунок 1.1 – Схема загального принципу роботи хаотичної карти

1.4 Порівняння основних технологій водяного знакування

Методи, що працюють у просторовій області, безпосередньо маніпулюють вихідними даними цифрового контенту – наприклад, значеннями пікселів у зображеннях або амплітудами сигналів у аудіо- та відеофайлах. Вони є найпростішими та найпоширенішими варіантами для вбудовування цифрових водяних знаків.

Таблиця 1.1 – Порівняння основних технологій водяного знакування

Характеристика	Методи в просторовій (часовій) області	Методи в частотній області
Принцип роботи	Пряма зміна пікселів/семплів	Перетворення (DCT, DFT, DWT), зміна коефіцієнтів
Простота реалізації	Простіші, обчислювально дешевші	Складніші через використання перетворень
Стійкість до обробки	Низька стійкість до стиснення, шуму, фільтрації	Вища стійкість до обробки; можливі вразливості до геометричних атак
Прозорість	Висока (наприклад, LSB), але обмежена стійкістю	Можливий добрий компроміс між стійкістю та прозорістю
Ємність	Може бути дуже високою (наприклад, LSB)	Обмежена через балансування прозорості та стійкості
Вразливість до атак	Висока; легко змінити чи видалити водяний знак	Менша, але існують спеціалізовані частотні атаки
Основне застосування	Стеганографія, прості маркування, коли стійкість не критична	Захист авторських прав, автентифікація, контроль розповсюдження контенту

Методи (таблиця 1.1), що працюють у просторовій області, безпосередньо маніпулюють вихідними даними цифрового контенту – наприклад, значеннями пікселів у зображеннях або амплітудами сигналів у аудіо- та відеофайлах. Вони є найпростішими та найпоширенішими варіантами для вбудовування цифрових водяних знаків.

Переваги методів просторової області:

- простота та інтуїтивність реалізації. Завдяки прямому втручанню у дані контенту ці методи легко розуміються й не вимагають складних математичних обчислень. Вони ідеально підходять для швидкої імплементації;

- низькі обчислювальні витрати. Оскільки немає потреби у перетвореннях або аналізі частотних характеристик сигналу, процес вбудовування і витягування водяного знаку виконується дуже швидко;

- висока ємність для вбудовування даних. Наприклад, метод найменш значущого біта (LSB) дозволяє вбудувати великий обсяг інформації без помітної зміни візуальної якості зображення чи аудіофайлу.

Недоліки методів просторової області:

- висока чутливість до змін контенту. Будь-яка базова обробка, така як стиснення, масштабування, обертання або редагування файлу, може легко пошкодити або знищити вбудований водяний знак;

- низька стійкість до атак. Оскільки дані змінюються прямо у відкритому вигляді, такі водяні знаки можуть бути легко виявлені та видалені або модифіковані за допомогою простих атак;

- обмежена надійність для захисту авторських прав. Через нестійкість до змін даних методи просторової області зазвичай не підходять для критичних задач автентифікації та захисту авторських прав.

Типові приклади:

- метод найменш значущого біта (LSB);

- адитивні методи (додавання невеликого шуму або модифікація значень);

- методи на основі аналізу текстур зображення.

Методи, що працюють у частотній області, базуються на перетворенні цифрового контенту в інший простір, де дані представлені як частотні компоненти. Після вбудовування водяного знаку в частотні коефіцієнти виконується зворотнє перетворення, щоб отримати кінцевий файл.

Переваги методів частотної області:

- підвищена стійкість до обробки. Стиснення, фільтрація, додавання шуму та інші стандартні форми обробки здебільшого впливають на високочастотні компоненти, тому водяні знаки, вбудовані в середньочастотні або низькочастотні зони, зберігаються краще;

- оптимальне використання властивостей людського зору (HVS). Завдяки точному налаштуванню частотних складових можна досягти такої інтеграції водяного знаку, яка залишається непомітною для людини, одночасно зберігаючи високу стійкість;

- підвищена стійкість до геометричних атак (у випадку DWT). Методи на основі дискретного вейвлет-перетворення (DWT) завдяки своїй багатороздільній природі краще протистоять таким атакам, як масштабування, обертання чи обрізка.

Недоліки методів частотної області:

- вища складність реалізації. Процес вимагає обчислювально інтенсивних перетворень (наприклад, DCT, DFT, DWT), що збільшує час обробки і потребує більших ресурсів;

- можливість специфічних атак на частотні компоненти. Деякі атаки спеціально спрямовані на маніпулювання або видалення водяних знаків, розташованих у частотній області;

- складність досягнення балансу між ємністю, стійкістю та прозорістю. При збільшенні ємності часто страждає прозорість або стійкість водяного знаку.

Типові приклади:

- вбудовування на основі дискретного косинусного перетворення (DCT);
- методи з використанням дискретного перетворення Фур'є (DFT);
- техніки на базі дискретного вейвлет-перетворення (DWT).

На сьогодні основний фокус досліджень у цій галузі спрямований на створення комбінованих або гібридних технологій, які намагаються об'єднати переваги обох підходів – просторового та частотного. Зокрема:

- активно розробляються адаптивні стратегії вбудовування, де водяний знак динамічно адаптується до локальних характеристик контенту для підвищення прозорості та стійкості;
- інтенсивно вивчаються методи, що забезпечують стійкість до геометричних атак – через використання інваріантних ознак або спеціального кодування знаків;
- набирають популярності рішення на базі штучного інтелекту та машинного навчання, які дозволяють автоматично оптимізувати процеси вбудовування й витягування водяних знаків залежно від типу даних і характеру атак.

Вибір конкретної технології водяного знакування завжди повинен базуватися на чітких вимогах до системи захисту: чи є пріоритетом максимальна стійкість до атак, висока прозорість для користувача чи велика ємність для прихованої інформації. Ідеального універсального рішення не існує, і в кожному випадку необхідний компроміс між обчислювальною складністю, стійкістю водяного знаку, його непомітністю та обсягом переданої інформації.

1.5 Класифікація цифрових водяних знаків

Цифрові водяні знаки можна класифікувати за різними ознаками, зокрема за методами вбудовування та рівнем захисту інформації. Нижче

наведено основні типи водяних знаків, які широко застосовуються у сучасних інформаційних технологіях:

1.Стеганографічні водяні знаки.

Стеганографічні водяні знаки розроблені для прихованого впровадження інформації всередину цифрового контенту таким чином, щоб сам факт існування водяного знаку був непомітним для сторонніх спостерігачів. Головна мета – забезпечити таємність та невидимість процесу передачі даних.

Інформація вбудовується у такі елементи файлу, як:

- окремі пікселі в цифрових зображеннях;
- бітові потоки аудіо- та відеофайлів;
- пробіли, символи або синтаксичні особливості у текстових документах.

Модифікації здійснюються настільки тонко, що вони не змінюють візуальне або аудіальне сприйняття контенту користувачем, залишаючи його якість незмінною на перший погляд.

Основні сфери застосування:

- прихована ідентифікація або маркування файлів для боротьби з несанкціонованим розповсюдженням;
- захист від копіювання і незаконного використання цифрового контенту;
- передача конфіденційних повідомлень у прихованій формі;
- підтвердження автентичності даних без прямого виявлення факту захисту.

2. Криптографічні водяні знаки.

Криптографічні водяні знаки орієнтовані на підвищення рівня безпеки інформації шляхом використання методів шифрування. Такі знаки не лише приховують дані, але й забезпечують їхню достовірність і захист від підробки.

У процесі вбудовування:

- водяний знак шифрується або підписується з використанням стійких криптографічних алгоритмів (наприклад, RSA, AES, або алгоритми цифрового підпису);

- інтеграція знаку в контент відбувається таким чином, що для його виявлення чи перевірки необхідно мати відповідний криптографічний ключ або цифровий сертифікат автентифікації;

- без наявності правильного ключа інформація, навіть якщо буде виявлена, залишиться незрозумілою для зловмисника.

Основні сфери застосування:

- захист авторських прав і підтвердження права власності на цифрові твори;

- підтвердження автентичності електронних документів у фінансовій, правовій та урядовій сферах;

- створення систем цифрового підпису та шифрування важливої інформації для запобігання її підробці чи несанкціонованій зміні.

3. Змішані (комбіновані) водяні знаки.

Змішані водяні знаки поєднують в собі властивості як стеганографічних, так і криптографічних підходів, що дозволяє створити багаторівневу систему захисту цифрового контенту.

Метод реалізації:

- спочатку інформацію, яка має бути захищена, шифрують за допомогою криптографічних алгоритмів для забезпечення її секретності та достовірності;

- після шифрування зашифрований водяний знак приховано вбудовується в цифровий об'єкт із використанням стеганографічних методів;

- таким чином, навіть якщо прихований водяний знак буде виявлений шляхом аналізу контенту, розшифрувати або використати його без відповідного ключа неможливо.

Основні сфери застосування:

- захист стратегічно важливої інформації, де важлива і конфіденційність, і цілісність даних;

- використання в медичних, військових та урядових системах для захисту критично важливих цифрових документів;

- у складних системах управління правами на контент (DRM-системах), де потрібна максимальна безпека та контроль над цифровими активами.

Класифікація цифрових водяних знаків за способом вбудовування і рівнем захисту допомагає краще зрозуміти, який саме тип знаку є оптимальним у конкретній ситуації.

Стеганографічні методи більше підходять для задач, де важлива непомітність передачі даних. Криптографічні методи ідеальні для тих випадків, коли необхідно гарантувати достовірність і захист від підробки. Змішані підходи пропонують компроміс між цими двома цілями, створюючи системи з максимальною стійкістю до атак.

2 ХАОТИСНІ КАРТИ ЯК ОСНОВА ДЛЯ ЗАХИСТУ ІНФОРМАЦІЇ

2.1 Хаотичні системи та карти

Використання хаотичних систем, зокрема так званих хаотичних карт, набуває дедалі більшого значення в галузі інформаційної безпеки. Ці системи мають низку властивостей – таких як складна структура поведінки, висока чутливість до початкових умов, непередбачуваність та псевдовипадковість, – що відкривають широкі можливості для створення сучасних засобів захисту даних і криптографічних протоколів.

Хаотичні системи – це особливий тип динамічних систем, які хоч і визначаються точними математичними правилами, демонструють поведінку, що на перший погляд здається випадковою або безладною. Їх головна особливість полягає в тому, що навіть незначні зміни на початковому етапі призводять до суттєвих відмінностей у подальшому розвитку системи. Цей феномен часто називають ефектом метелика.

До основних характеристик таких систем належать:

- детермінованість – розвиток системи повністю визначається її початковими умовами і правилами, за якими вона працює. Випадкові чинники відсутні, хоча поведінка виглядає хаотичною;

- нелінійність – залежності між змінними системи не є простими (лінійними), а мають складний характер, що призводить до виникнення складних динамічних процесів;

- висока чутливість до початкових умов – навіть малі зміни на старті системи можуть призвести до суттєво інших результатів з часом, що суттєво ускладнює прогнозування її поведінки на довгих проміжках;

- топологічне перемішування – у процесі еволюції система здатна так "перемішати" точки у своєму просторі станів, що будь-які два набори значень зрештою значно змінять своє розташування відносно один одного;

- наявність великої кількості нестійких періодичних траєкторій – система містить нескінченну множину циклічних режимів, але вони нестійкі і легко порушуються.

Хаотичні карти – це дискретні моделі (тобто такі, що працюють на послідовності окремих кроків), які демонструють такі ж властивості, як і безперервні хаотичні системи. Вони дозволяють генерувати послідовності значень, що мають ознаки хаосу, і саме це робить їх корисними для застосувань у сфері захисту інформації.

Існує багато різновидів хаотичних карт, які використовуються для генерації складних послідовностей. Серед найбільш відомих можна виділити:

- логістична карта – одна з найпростіших одновимірних систем, яка демонструє перехід від регулярної поведінки до хаотичної залежно від параметрів;

- карта Ено – двовимірна система, яка генерує складні двовимірні траєкторії та широко використовується для моделювання складних динамічних процесів;

- карта пекаря – модель, яка імітує процес "розтягування і складання" простору, подібно до того, як замішується тісто;

- карта-намет (tent map) – ще один приклад простої одновимірної системи, здатної швидко переходити до хаотичного режиму при певних параметрах.

Особливості хаотичних карт дозволяють ефективно застосовувати їх у різних аспектах захисту інформації:

- генерація псевдовипадкових чисел. Хаотичні карти можуть використовуватись для створення на вигляд випадкових послідовностей чисел. Такі послідовності мають застосування у шифрувальних алгоритмах – наприклад, як ключові потоки у потокових шифрах чи для генерації ініціалізаційних векторів у блокових шифрах;

- шифрування даних. За допомогою хаотичних карт можна створювати криптосистеми, які здійснюють перемішування і заміну даних на основі

ітерацій карти. Завдяки високій чутливості до початкових параметрів, навіть мінімальна помилка у введених значеннях призведе до неможливості розшифрувати дані, що підвищує криптостійкість;

- цифрове водяне знакування. Послідовності, отримані з хаотичних карт, можна використовувати для створення водяних знаків у мультимедійних файлах. Це забезпечує високу стійкість водяного знака до спроб його видалення або до змін контенту (наприклад, стиснення, обрізання);

- стеганографія. Хаотичні карти дозволяють визначати, в яких місцях цифрового контенту приховувати інформацію. Такий підхід робить стеганографічне повідомлення важким для виявлення, оскільки місця приховування обираються псевдовипадковим і складним чином.

Завдяки своїм особливим властивостям, хаотичні карти стають потужним інструментом для розробки новітніх методів криптографії, стеганографії та цифрового водяного знакування. У подальших розділах ми детально розглянемо конкретні приклади їх практичного застосування в кожній із зазначених сфер.

2.2 Властивості хаотичних карт

Хаотичні карти мають низку фундаментальних властивостей, завдяки яким вони стають привабливим інструментом для задач цифрового водяного знакування. Дві основні характеристики – ергодичність і чутливість до початкових умов – відіграють ключову роль у забезпеченні стійкості, безпеки та ефективності водяних знаків.

Ергодичність означає, що коли система еволюціонує достатньо довго, її траєкторія поступово проходить майже через всі можливі стани, доступні у фазовому просторі. Інакше кажучи, хаотична послідовність не застряє у якійсь вузькій ділянці простору, а рівномірно заповнює всю доступну область. Це дозволяє очікувати, що середнє значення величини, обчислене вздовж однієї довгої траєкторії (часове середнє), буде близьким до

середнього значення, отриманого по всіх можливих станах системи (просторове середнє).

Практичне значення для водяного знакування:

- рівномірне покриття контенту. Завдяки ергодичності, елементи хаотичної послідовності охоплюють різні ділянки цифрового контенту – зображення, аудіо або відео. Якщо така послідовність використовується для вбудовування водяного знака, енергія водяного знака рівномірно розподіляється по всьому контенту. Це підвищує стійкість до атак, які спрямовані на локальне пошкодження даних (наприклад, вирізання фрагментів зображення або часткове стискання);

- передбачуваність статистичних властивостей. Ергодичність дозволяє аналізувати статистику хаотичних послідовностей (середнє значення, дисперсія, автокореляція тощо). Ці властивості можуть бути використані під час розробки алгоритмів для виявлення або перевірки наявності водяного знака у контенті. Наприклад, якщо водяний знак накладає специфічну статистичну «підписку», це можна виявити навіть після спотворення контенту;

- псевдовипадковість. Ергодична поведінка створює ілюзію випадковості у послідовності значень, що генерується картою. Це важливо для захисту водяного знака: псевдовипадкова послідовність складно передбачувана для зловмисника і виглядає як шум, що ускладнює її виявлення або підробку.

Чутливість до початкових умов, ця властивість, відома також як ефект метелика, означає, що навіть мікроскопічна зміна у початкових умовах призводить до значної різниці у траєкторіях системи вже після кількох ітерацій. Дві майже ідентичні початкові точки призводять до абсолютно різних послідовностей значень.

Чутливість до початкових умов вимірюється за допомогою показника Ляпунова. Якщо цей показник додатний, це свідчить про наявність хаотичної поведінки і експоненційне розходження траєкторій.

Практичне значення для водяного знакування:

- стійкість секретного ключа. У водяних знаках, де початкові умови хаотичної карти використовуються як секретний ключ, ця властивість гарантує безпеку. Якщо зловмисник спробує підібрати ключ і навіть наблизиться дуже близько до справжнього значення, згенерована ним послідовність все одно буде абсолютно іншою. Це практично унеможливлює підбір правильного шаблону або відтворення водяного знака;

- захист від підробки. Навіть мінімальна помилка у відтворенні ключа (початкових параметрів) призведе до формування зовсім іншого водяного знака. Це ускладнює створення фальшивих водяних знаків, які можна було б видати за справжні;

- генерація складних шаблонів. Завдяки швидкому розходженню траєкторій можна створювати довгі, складні та малопередбачувані послідовності, які ідеально підходять для вбудовування інформації у контент без повторів або шаблонів, які легко аналізувати.

Комбінація цих двох властивостей робить хаотичні карти надзвичайно ефективними для задач водяного знакування:

- ергодичність забезпечує рівномірність і прогнозованість статистичних властивостей послідовностей. Це важливо для контролю за розподілом водяного знака у контенті і правильного налаштування алгоритмів виявлення;

- чутливість до початкових умов гарантує криптографічну стійкість і унеможливлює відтворення водяного знака без точного знання секретного ключа.

Разом ці властивості дозволяють розробляти системи цифрового водяного знакування, які є водночас надійними, малопомітними і захищеними від атак.

2.3 Популярні хаотичні карти

Одним із найбільш відомих прикладів хаотичних систем є логістична карта. Вона вирізняється винятковою простотою математичної моделі, що дозволяє відображати значення в межах від нуля до одиниці на самих себе. Залежно від обраного параметра ця карта може демонструвати як передбачувану поведінку, так і складну хаотичну динаміку. Коли параметр досягає певного порогу, система переходить у стан повного хаосу, у якому траєкторії значень стають непередбачуваними і рівномірно заповнюють увесь доступний простір. Саме в цьому режимі логістична карта демонструє ергодичність і позитивний показник Ляпунова – ознаки наявності хаосу. Саме завдяки цим властивостям вона часто застосовується для створення псевдовипадкових послідовностей у сфері цифрового водяного знакування. Її здатність швидко розходитися навіть при незначних змінах початкових умов робить її придатною для генерації ключів, які складно підробити чи відтворити без точного знання параметрів.

Ще однією популярною хаотичною системою є tent-карта, яка отримала свою назву через характерний вигляд графіка, що нагадує намет. Вона також працює в межах інтервалу від нуля до одиниці та проявляє хаотичну поведінку при певних значеннях керуючого параметра. Її динаміка, хоч і простіша за формою, дає змогу отримувати послідовності зі схожими статистичними властивостями, як і в логістичної карти. Лінійні сегменти цієї карти можуть надавати певні переваги при математичному аналізі і спрощують оцінку її властивостей. Завдяки здатності генерувати складні та рівномірно розподілені послідовності, tent-карта також широко використовується для створення водяних знаків та криптографічних ключів.

Окрім цих класичних прикладів, значного поширення набули генералізовані хаотичні карти. Вони є результатом розширення або ускладнення вже відомих карт шляхом введення додаткових параметрів, нових функціональних залежностей або навіть переходу до багатовимірних

систем. Такі модифікації дозволяють суттєво розширити фазовий простір, зробити поведінку системи ще менш передбачуваною, а також поліпшити статистичні характеристики згенерованих послідовностей. Наприклад, введення нових параметрів у логістичну карту може призвести до ще складнішої динаміки, що підвищує безпеку генерованих ключів. Багатовимірні системи, які складаються з декількох взаємозалежних хаотичних карт, дозволяють формувати більш об’ємний простір ключів, ускладнюючи задачу атакуючому. Окрему увагу привертають карти, побудовані на основі кусково-лінійних функцій або тригонометричних залежностей. Такі моделі забезпечують гнучкість налаштування властивостей хаотичних послідовностей під конкретні завдання.

Для систем водяного знакування генералізовані карти відкривають нові можливості. Вони дозволяють створювати більш надійні схеми, здатні витримувати криптоаналітичні та статистичні атаки, яким прості карти можуть бути вразливі. Завдяки своїй підвищеній складності ці карти формують водяні знаки, які важче виявити, підробити чи видалити. Крім того, гнучкість керування параметрами дає змогу точніше адаптувати динаміку системи під конкретні вимоги: від забезпечення високої стійкості до атак до оптимізації швидкості обчислень.

Вибір конкретної хаотичної карти завжди залежить від цілей, які ставляться перед системою водяного знакування. В одних випадках на перший план виходить безпека і криптостійкість, в інших – ефективність та простота реалізації. Слід зазначити, що ця галузь постійно розвивається: дослідники створюють усе новіші модифікації хаотичних карт, прагнучи досягти ще більшої ефективності в захисті цифрової інформації.

2.4 Використання хаотичних карт у криптографії та стеганографії

Хаотичні карти, завдяки своїй здатності породжувати складні й на перший погляд випадкові послідовності, викликають значний інтерес у сфері

захисту інформації. Їхні основні властивості – чутливість до початкових умов, ергодичність і непередбачувані траєкторії – формують підґрунтя для створення нових криптографічних алгоритмів. Це дозволяє не лише ускладнювати аналіз системи для потенційного атакувальника, але й забезпечувати високий ступінь безпеки завдяки мінімальній залежності між схожими початковими налаштуваннями.

Одним із найбільш поширених застосувань хаотичних карт є генерація послідовностей, які наближені до випадкових. Такі послідовності можна ефективно використовувати як генератори псевдовипадкових чисел. У цьому випадку початкові умови та параметри карти виконують роль секретного ключа. Навіть незначна зміна значення цього ключа здатна радикально змінити вихідні послідовності, що критично важливо для безпеки криптографічних систем. Ці псевдовипадкові послідовності застосовуються в різних криптографічних процесах: від генерації ключових потоків у потокових шифрах до ініціалізації векторів у блокових шифрах.

Крім того, хаос відіграє важливу роль у побудові як потокових, так і блокових шифрів. В потокових шифрах псевдовипадкова послідовність, яка генерується хаотичною картою, поєднується з відкритим текстом шляхом використання бітових операцій (зазвичай XOR), що дозволяє отримати шифротекст. Завдяки тому, що ключова послідовність повністю визначається початковими умовами, процес розшифрування є дзеркальним до шифрування. У випадку блокових шифрів ітерації хаотичних карт можуть визначати порядок перестановок і підстановок у даних – ці операції формують основу для змішування і підстановки інформації на рівні бітів чи блоків. Оскільки найменша зміна ключа змінює весь хід таких операцій, це значно ускладнює будь-які спроби криптоаналізу.

Однак незважаючи на ці переваги, використання хаотичних карт у шифруванні вимагає уважного підходу. Питання дискретизації, пов’язані з кінцевою точністю числових обчислень, можуть порушити хаотичний характер карти в цифрових реалізаціях. Тому розробникам доводиться

ретельно аналізувати періодичність, лінійність і стійкість до атак на запропоновані алгоритми.

Ще однією важливою сферою застосування є генерація криптографічних ключів. Ідея полягає в тому, щоб початкові умови, отримані з випадкового джерела або за допомогою хеш-функції, пропустити через хаотичну карту. Після певної кількості ітерацій вихідні значення можуть бути використані як основа для формування складних і непередбачуваних ключів потрібної довжини.

Також хаотичні карти можна інтегрувати в побудову криптографічних хеш-функцій. У цьому випадку повідомлення, яке хешується, визначає початковий стан або параметри карти. Після кількох ітерацій кінцевий стан системи виступає як хеш-значення. Завдяки чутливості до початкових умов, навіть незначна зміна вхідних даних призводить до суттєвої зміни хешу, що забезпечує ефект лавини – одну з критичних властивостей надійних хеш-функцій.

Не менш перспективним є застосування хаотичних карт у стеганографії – галузі, яка займається прихованим передаванням інформації в зовні нешкідливих цифрових об’єктах, таких як зображення, аудіо чи відео. Завдяки своїй здатності генерувати складні й малопередбачувані послідовності, хаотичні карти дозволяють підвищити як рівень безпеки таких методів, так і їхню непомітність.

Одним із найбільш поширених прийомів є використання хаотичних карт для вибору місць, де саме вбудовується прихована інформація. Індекси пікселів, аудіосемплів або інших елементів носія визначаються псевдовипадковою послідовністю, що залежить від секретного ключа. Це робить розміщення інформації нерегулярним і малопомітним для стороннього спостерігача, який не володіє цим ключем.

Деякі методи також комбінують секретне повідомлення з бітами хаотичної послідовності перед самим вбудовуванням. Це ускладнює спроби

витягти приховану інформацію навіть у разі її виявлення, оскільки повідомлення саме по собі додатково маскується.

Ще однією цікавою можливістю є формування стійких до виявлення шаблонів для приховання інформації. Замість простого послідовного або регулярного вбудовування даних, хаотичні карти дозволяють створювати складні структури, які візуально та статистично виглядають як природна частина носія. Це значно ускладнює застосування методів статистичного аналізу, які часто використовуються для виявлення стеганографічного вмісту.

Також хаотичні послідовності можуть брати участь у динамічному регулюванні процесу вбудовування. Наприклад, в залежності від локальних властивостей носія (яскравість пікселя, інтенсивність звуку тощо), хаотична карта може визначати, наскільки сильно або слабо змінювати той чи інший елемент. Це дозволяє краще приховувати інформацію в областях, де такі зміни найменш помітні.

Таким чином, використання хаотичних карт у стеганографії забезпечує додаткові переваги: підвищену безпеку прихованих даних, кращу непомітність змін у носії, а також стійкість до аналізу з боку сторонніх. Водночас, як і в криптографії, застосування таких методів вимагає уважного аналізу й тестування, щоб уникнути втрат хаотичних властивостей у цифровій реалізації.

Хаотичні карти відкривають широкі можливості для створення сучасних засобів захисту інформації як у криптографії, так і в стеганографії. Їхні унікальні властивості дозволяють формувати складні, надійні та важкопрогнозовані алгоритми, здатні витримувати сучасні методи аналізу та атак. Однак для їх ефективного використання необхідно ретельно враховувати особливості цифрових реалізацій – насамперед вплив дискретизації й обмеженої точності обчислень. Подальші дослідження та експериментальні розробки в цьому напрямку можуть привести до появи ще більш досконалих і стійких до атак систем інформаційної безпеки.

2.5 Використання хаотичних карт у цифровому водяному маркуванні

Як хаотичні карти можуть бути використані для генерації ключів або вбудовування водяних знаків:

1. Генерація криптографічних ключів за допомогою хаотичних карт (рисунок 2.1).

Хаотичні карти можуть слугувати ефективним інструментом для створення криптографічних ключів завдяки своїм фундаментальним властивостям – непередбачуваності, чутливості до початкових умов та складній динаміці. Ці характеристики дозволяють генерувати довгі та складні послідовності чисел, які важко відтворити або передбачити без знання початкових параметрів. Процес генерації ключів за допомогою хаотичних карт може виглядати так:

- вибір початкових умов і параметрів карти – ці значення виступають як секретний ключ, який визначає весь подальший хід генерації;
- ітерація хаотичної карти – багаторазове застосування карти для отримання послідовності чисел;
- перетворення послідовності в криптографічний ключ – отримані значення можуть бути безпосередньо використані як ключ або додатково оброблені (наприклад, хешовані або перетворені у бітові послідовності відповідної довжини).

Такі ключі можуть застосовуватися в різних криптографічних протоколах:

- як ключі симетричного шифрування (AES);
- як сесійні ключі для протоколів безпечної комунікації (TLS);
- для ініціалізації генераторів псевдовипадкових чисел або хеш-функцій.

Перевага такого підходу полягає в тому, що навіть мінімальна зміна початкових умов генерує зовсім іншу послідовність, що ускладнює підбір ключа методом перебору.

2.Вбудовування цифрових водяних знаків за допомогою хаотичних карт.

Цифровий водяний знак – це спеціальна інформація, прихована всередині цифрових файлів (зображень, аудіо, відео тощо) з метою підтвердження авторства, автентичності або для захисту від несанкціонованого копіювання. Хаотичні карти дозволяють значно підвищити надійність і непомітність водяних знаків.

Основна ідея полягає в тому, що хаотична карта визначає місця і спосіб вбудовування водяного знаку у файл таким чином, щоб цей процес був практично невідтворюваний без знання ключа.

Процес може складатися з таких етапів:

- генерація хаотичної послідовності індексів. На основі заданих початкових умов карта генерує послідовність, що вказує, в які конкретно пікселі (для зображення) або семпли (для аудіо) буде вбудована інформація;
- шифрування водяного знаку. Перед вбудовуванням сам водяний знак може бути шифрований або перетворений за допомогою тієї ж хаотичної послідовності. Це додає ще один рівень захисту – навіть якщо приховану інформацію буде знайдено, без ключа вона залишиться незрозумілою;
- вбудовування водяного знаку. Біти водяного знаку інтегруються у вибрані місця файлу, часто шляхом незначної модифікації значень пікселів або семплів (наприклад, зміною молодшого біта);
- вилучення водяного знаку. Для вилучення водяного знаку необхідно повторно згенерувати ту ж саму хаотичну послідовність на основі початкового ключа, що дозволить точно визначити місця приховання та відновити інформацію.

Переваги використання хаотичних карт у водяних знаках:

- висока стійкість до атак – завдяки чутливості до ключа сторонній спостерігач не може просто так знайти місця вбудовування або видалити знак;

- непомітність – псевдовипадковий вибір місць для інтеграції зменшує можливість виявлення за допомогою статистичного аналізу;
- захист авторських прав і цілісність – надійне вбудовування водяного знаку дозволяє підтверджувати автентичність цифрових творів навіть після їх розповсюдження.

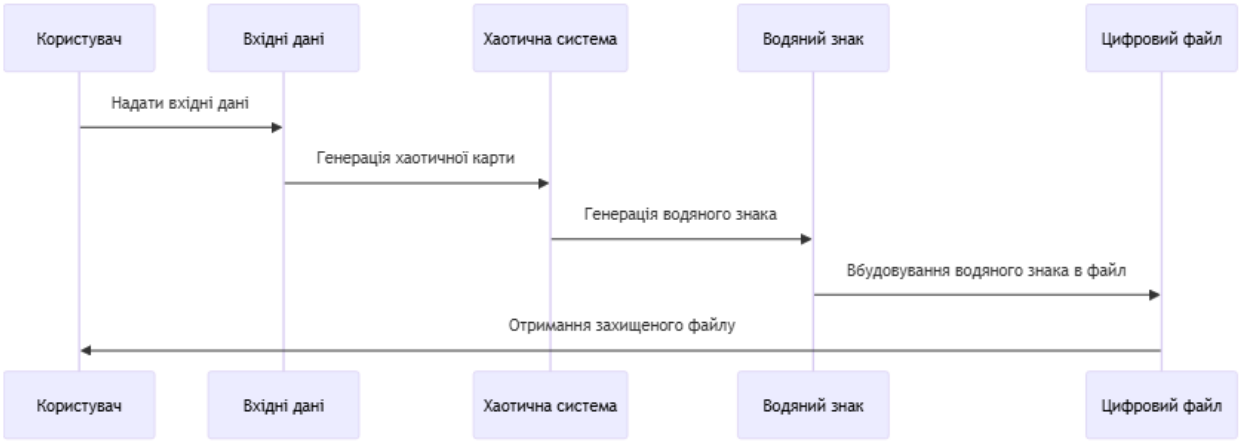

Рисунок 2.1 – Алгоритм генерації хаотичної карти

Хаотичні карти, завдяки своїм унікальним математичним властивостям, відкривають широкі можливості як у криптографії, так і в цифровій стеганографії. Вони дозволяють створювати стійкі, непередбачувані ключі та надійно інтегрувати водяні знаки у цифрові файли. Однак для практичного застосування необхідно враховувати потенційні проблеми – вплив дискретизації, обмеження точності обчислень і необхідність ретельного криптоаналізу – аби гарантувати надійність і безпеку таких систем.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

У цьому розділі досліджено програмну реалізацію системи нанесення цифрових водяних знаків із застосуванням хаотичних карт. Програма втілена мовою Python з використанням бібліотек OpenCV та NumPy. Головна задача програми полягає в тому, щоб забезпечити надійне впровадження водяного знака в зображення, його стійкість до атак та можливість відновлення інформації. У розділі розкрито структуру програми, її архітектуру, ключові функціональні компоненти, а також проведено тестування і оцінку сильних та слабких сторін.

3.1 Структура програми та архітектура

Програма для нанесення цифрових водяних знаків із використанням хаотичних карт побудована за модульним принципом, що забезпечує її гнучкість, масштабованість та зручність у підтримці. Основна архітектура програми складається з кількох функціональних блоків, кожен із яких виконує окрему задачу. Взаємодія між блоками організована таким чином, щоб забезпечити послідовність виконання етапів нанесення водяного знака, його витяг та оцінку стійкості до атак.

Розглянемо основні компоненти програми:

Головний модуль (main.py) є точкою входу в програму. Він забезпечує взаємодію користувача з програмою через інтерфейс командного рядка або графічний інтерфейс (за потреби). У цьому модулі викликаються всі основні функції, що відповідають за генерацію водяного знака, його впровадження, витяг, а також тестування стійкості до атак. Головний модуль також відповідає за обробку параметрів, які передаються користувачем, таких як шлях до зображення, текст водяного знака, параметри хаотичної карти тощо.

Блок для генерації хаотичної послідовності реалізований у функції logistic_map. Він генерує хаотичну послідовність чисел на основі логістичної карти, яка є основою для модифікації DCT-коефіцієнтів. Логістична карта визначається параметром r (рівень хаотичності) та початковим значенням seed. Ця послідовність забезпечує унікальність водяного знака та підвищує його стійкість до атак.

Блок обробки зображень включає функції apply_dct та apply_idct, які виконують пряме та зворотне дискретне косинусне перетворення (DCT/IDCT). Ці операції дозволяють працювати із частотними компонентами зображення, що є основою для впровадження водяного знака. DCT дозволяє розділити зображення на низько-, середньо- та високочастотні компоненти, що дає змогу впроваджувати водяний знак у найбільш стійкі до атак області.

Функція generate_text_watermark створює текстовий водяний знак у вигляді чорно-білого зображення. Користувач може налаштувати текст, розмір шрифту, товщину ліній та розмір зображення водяного знака. Цей блок дозволяє створювати водяні знаки різної складності, що підвищує їхню унікальність.

Функція embed_watermark виконує інтеграцію водяного знака в зображення-хост. Для цього використовується DCT-домен, де модифікуються низькочастотні компоненти зображення за допомогою хаотичної послідовності. Параметр alpha визначає інтенсивність впровадження водяного знака, що дозволяє балансувати між якістю зображення та стійкістю водяного знака.

Функція extract_watermark дозволяє відновити водяний знак із зображення. Вона використовує різницю між DCT-коефіцієнтами оригінального зображення та зображення з водяним знаком. Цей процес є зворотним до впровадження водяного знака.

Для перевірки стійкості водяного знака програма підтримує кілька типів атак:

- JPEG-стиснення (simulate_jpeg_compression) – моделює втрати якості через стиснення;
- додавання шуму (add_noise) – додає гаусівський шум до зображення;
- фільтрація (apply_filter) – застосовує медіанний фільтр для згладжування зображення. Ці атаки дозволяють оцінити, наскільки водяний знак зберігається після обробки зображення.

Функція evaluate_quality розраховує метрики PSNR (Peak Signal-to-Noise Ratio) та SSIM (Structural Similarity Index) для оцінки якості зображення після впровадження водяного знака. Ці метрики дозволяють визначити ступінь спотворення зображення та ефективність впровадження водяного знака.

Взаємодія між компонентами програми організована таким чином, щоб забезпечити послідовність виконання всіх етапів. Нижче наведено схему загальної архітектури програми.

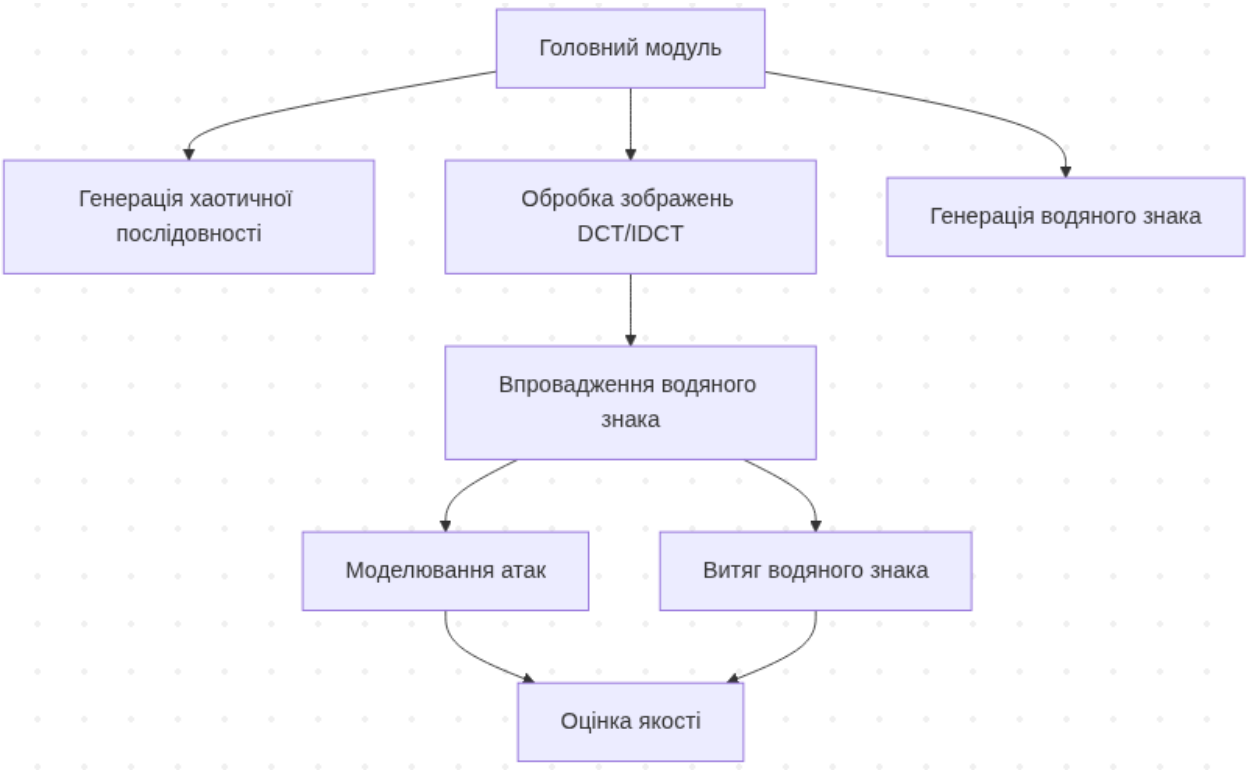

Рисунок 3.1 – Організація компонентів програми

Для кращого розуміння взаємодії функцій у програмі наведено схему викликів функцій.

З переваг такої архітектури можна визначити:

- модульність – кожен функціональний блок реалізований у вигляді окремої функції або модуля, що спрощує тестування, налагодження та розширення програми;
- гнучкість – програма дозволяє легко змінювати параметри, такі як текст водяного знака, параметри хаотичної карти, інтенсивність впровадження тощо;
- масштабованість – завдяки модульній структурі можна додавати нові типи атак або методи впровадження водяного знака без значних змін у коді.

Ця структура забезпечує ефективну реалізацію алгоритму нанесення цифрових водяних знаків із використанням хаотичних карт, а також дозволяє легко адаптувати програму до нових вимог.

3.2 Генерація хаотичної послідовності

Генерація хаотичної послідовності є важливим етапом у процесі нанесення цифрового водяного знака. Цей блок реалізований у функції logistic_map, яка генерує послідовність чисел на основі логістичної карти. Логістична карта є математичною моделлю, яка демонструє хаотичну поведінку при певних значеннях параметра. Формула логістичної карти має вигляд:

$$x(n+1) = r * x(n) * (1 - x(n)) \quad , \tag{3.1}$$

де: x(n) – поточне значення послідовності, r – параметр хаотичності (зазвичай у межах [3.57, 4]), x(0) – початкове значення (seed), яке визначає початок послідовності.

Ця послідовність використовується для модифікації DCT-коефіцієнтів під час впровадження водяного знака, забезпечуючи унікальність і стійкість до атак.

Алгоритм роботи функції logistic_map (лістинг 3.1):

- ініціалізується початкове значення x(0) (seed) та параметр r;
- генерується послідовність чисел заданої довжини n;
- кожне значення обчислюється за формулою логістичної карти;
- результуюча послідовність нормалізується (за потреби) для використання в алгоритмі впровадження водяного знака.

Лістинг 3.1 – Генерація хаотичної послідовності

```
def logistic_map(seed, r, n):
   # Ініціалізація послідовності
    sequence = []
    x = seed
    # Генерація послідовності
    for _ in range(n):
        x = r * x * (1 - x)  # Формула логістичної карти
        sequence.append(x)
    return sequence
```

Функція для генерації хаотичної послідовності приймає три вхідні параметри: seed, r та n. Параметр seed визначає початкове значення, з якого починається послідовність, і впливає на її унікальність. Параметр r є коефіцієнтом хаотичності, який визначає поведінку послідовності, а параметр n задає кількість значень, які потрібно згенерувати.

На етапі ініціалізації початкове значення x встановлюється рівним seed, а також створюється порожній список sequence, у який будуть зберігатися результати обчислень. Далі запускається цикл генерації, у якому для кожної ітерації обчислюється нове значення x за формулою логістичної карти. Кожне обчислене значення додається до списку sequence.

Після завершення циклу функція повертає список із n значень, які можна використовувати для модифікації DCT-коефіцієнтів під час впровадження водяного знака. Ця послідовність є детермінованою, тобто при

однакових параметрах seed і r завжди генерується однаковий результат, що забезпечує повторюваність у процесі впровадження та витягу водяного знака.

Приклад результату – для параметрів seed = 0.5, r = 3.9, n = 10 результатом буде послідовність, наприклад: [0.975, 0.0950625, 0.335499922265625, 0.8694649252599205, ...]

Ця послідовність є хаотичною, але детермінованою, тобто при однакових параметрах завжди генерується однаковий результат. Це забезпечує повторюваність у процесі впровадження та витягу водяного знака.

3.3 Обробка зображень

Обробка зображень є ключовим етапом у процесі нанесення цифрового водяного знака. Для роботи з частотними компонентами зображення використовуються операції дискретного косинусного перетворення (DCT) та його зворотного перетворення (IDCT). Ці операції дозволяють перейти від просторового представлення зображення до частотного, що є основою для впровадження водяного знака.

Дискретне косинусне перетворення розділяє зображення на низько-, середньо- та високочастотні компоненти. Водяний знак зазвичай впроваджується в низькочастотні області, оскільки вони є найбільш стійкими до атак, таких як стиснення або фільтрація. Зворотне перетворення (IDCT) дозволяє відновити зображення після модифікації його частотних компонентів.

Основні етапи обробки зображень:

- пряме DCT – перетворення зображення з просторового домену в частотний та розрахунок DCT-коефіцієнтів, які представляють енергію зображення в різних частотних діапазонах;

- модифікація DCT-коефіцієнтів – впровадження водяного знака шляхом зміни певних частотних компонентів;

- зворотне IDCT – перетворення модифікованих DCT-коефіцієнтів назад у просторовий домен для отримання зображення з водяним знаком.

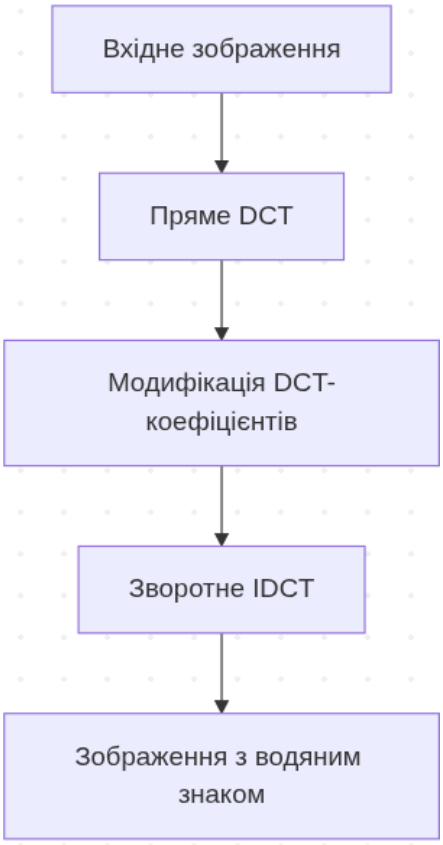

Рисунок 3.2 – Схема роботи блоку обробки зображень

Лістинг 3.2 – Реалізація функцій DCT та IDCT

```
import cv2
import numpy as np
def apply_dct(image):
    # Перетворення зображення у float32 для точності обчислень
    image_float = np.float32(image)
    # Виконання DCT
    dct_coefficients = cv2.dct(image_float)
    return dct_coefficients
def apply_idct(dct_coefficients):
   # Виконання IDCT
    restored_image = cv2.idct(dct_coefficients)
    # Перетворення результату у формат uint8
    return np.uint8(np.clip(restored_image, 0, 255))
```

Функція apply_dct (лістинг 3.2) приймає вхідне зображення у відтінках сірого, яке перед обробкою перетворюється у формат float32. Це забезпечує

високу точність обчислень під час виконання дискретного косинусного перетворення (DCT). Для виконання DCT використовується функція cv2.dct із бібліотеки OpenCV, яка обчислює частотні компоненти зображення. Результатом роботи функції є матриця DCT-коефіцієнтів, яка представляє енергію зображення в різних частотних діапазонах.

Функція apply_idct приймає матрицю DCT-коефіцієнтів і виконує зворотне дискретне косинусне перетворення (IDCT) за допомогою функції cv2.idct. Це дозволяє відновити зображення у просторовому домені. Після виконання IDCT результат перетворюється у формат uint8, щоб його можна було зберегти або відобразити у вигляді зображення.

Для зображення розміром 256x256 пікселів функція apply_dct повертає матрицю DCT-коефіцієнтів того ж розміру. Після виконання apply_idct відновлене зображення буде візуально ідентичним оригінальному, якщо DCT-коефіцієнти не були змінені.

3.4 Генерація текстового водяного знака

Генерація текстового водяного знака є важливим етапом у процесі нанесення цифрового водяного знака. Цей блок реалізований у функції generate_text_watermark, яка створює зображення водяного знака на основі заданого тексту. Водяний знак генерується у вигляді чорно-білого зображення, яке може бути інтегроване в зображення-хост.

Функція дозволяє налаштовувати такі параметри:

- розмір зображення водяного знака (ширина та висота);
- масштаб шрифту для тексту;
- товщина ліній тексту;
- розташування тексту в межах зображення.

Цей підхід забезпечує гнучкість у створенні водяних знаків різної складності та розміру, що дозволяє адаптувати їх до різних типів зображень.

Основні етапи генерації текстового водяного знака (рис. 3.3):

- створення порожнього зображення заданого розміру (чорно-білого);
- визначення параметрів тексту (шрифт, масштаб, товщина);
- розрахунок позиції тексту для його центрування;
- нанесення тексту на зображення за допомогою функції OpenCV.

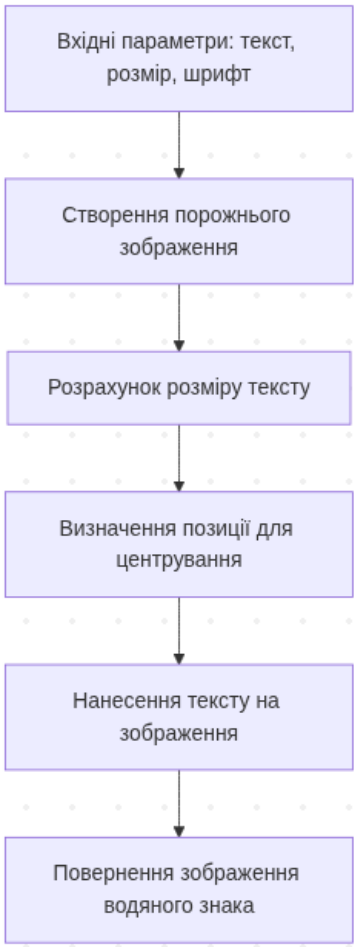

Рисунок 3.3 – Схема роботи функції генерації текстового водяного знака

Лістинг 3.3 – Генерація текстового водяного знака

```
import cv2
import numpy as np
def generate_text_watermark(text, width, height, font_scale=1,
thickness=2):
    # Створення порожнього зображення (чорно-білого)
    watermark = np.zeros((height, width), dtype=np.uint8)
    # Вибір шрифту
    font = cv2.FONT_HERSHEY_SIMPLEX
    # Розрахунок розміру тексту
    text_size = cv2.getTextSize(text, font, font_scale,
thickness)[0]
    text_width, text_height = text_size
```

```
    # Визначення позиції тексту для центрування
    text_x = (width - text_width) // 2
    text_y = (height + text_height) // 2
    # Нанесення тексту на зображення
    cv2.putText(watermark, text, (text_x, text_y), font,
font_scale, 255, thickness)
    return watermark
```

Функція для генерації текстового водяного знака приймає кілька вхідних параметрів (лістинг 3.3): text – текст, який буде використаний як водяний знак, width та height – розміри зображення водяного знака, font_scale – масштаб шрифту, що визначає розмір тексту, та thickness – товщину ліній тексту. Ці параметри дозволяють налаштувати вигляд водяного знака відповідно до потреб користувача.

На першому етапі створюється порожнє чорно-біле зображення заданого розміру за допомогою функції np.zeros. Це зображення слугує основою для нанесення тексту. Далі за допомогою функції cv2.getTextSize визначаються розміри тексту для заданих параметрів шрифту, масштабу та товщини. Це дозволяє розрахувати позицію тексту таким чином, щоб він був розташований по центру зображення.

Після цього текст наноситься на зображення за допомогою функції cv2.putText. Колір тексту встановлюється як білий (значення 255), що забезпечує чіткий контраст із чорним фоном. У результаті функція повертає зображення водяного знака у вигляді матриці numpy.ndarray, яке можна використовувати для подальшого впровадження у зображення-хост.

Для параметрів text = "Watermark", width = 300, height = 100, font_scale = 2, thickness = 3 результатом буде зображення розміром 300x100 пікселів із текстом "Watermark", розташованим по центру.

3.5 Впровадження водяного знака

Впровадження водяного знака є ключовим етапом у процесі нанесення цифрового водяного знака. Цей блок реалізований у функції

embed_watermark, яка виконує інтеграцію водяного знака в зображення-хост. Для цього використовується дискретне косинусне перетворення (DCT), яке дозволяє працювати з частотними компонентами зображення.

Основна ідея полягає в тому, щоб модифікувати низькочастотні компоненти DCT-коефіцієнтів зображення-хоста за допомогою значень водяного знака та хаотичної послідовності. Це забезпечує стійкість водяного знака до атак, таких як стиснення JPEG або додавання шуму.

Основні етапи впровадження водяного знака (рис. 3.4):

- виконання DCT для зображення-хоста;

- модифікація низькочастотних компонентів DCT-коефіцієнтів – значення водяного знака (0 або 1) впливають на зміну відповідних компонентів та хаотична послідовність використовується для додаткової унікальності;

- виконання зворотного DCT (IDCT) для отримання зображення з водяним знаком.

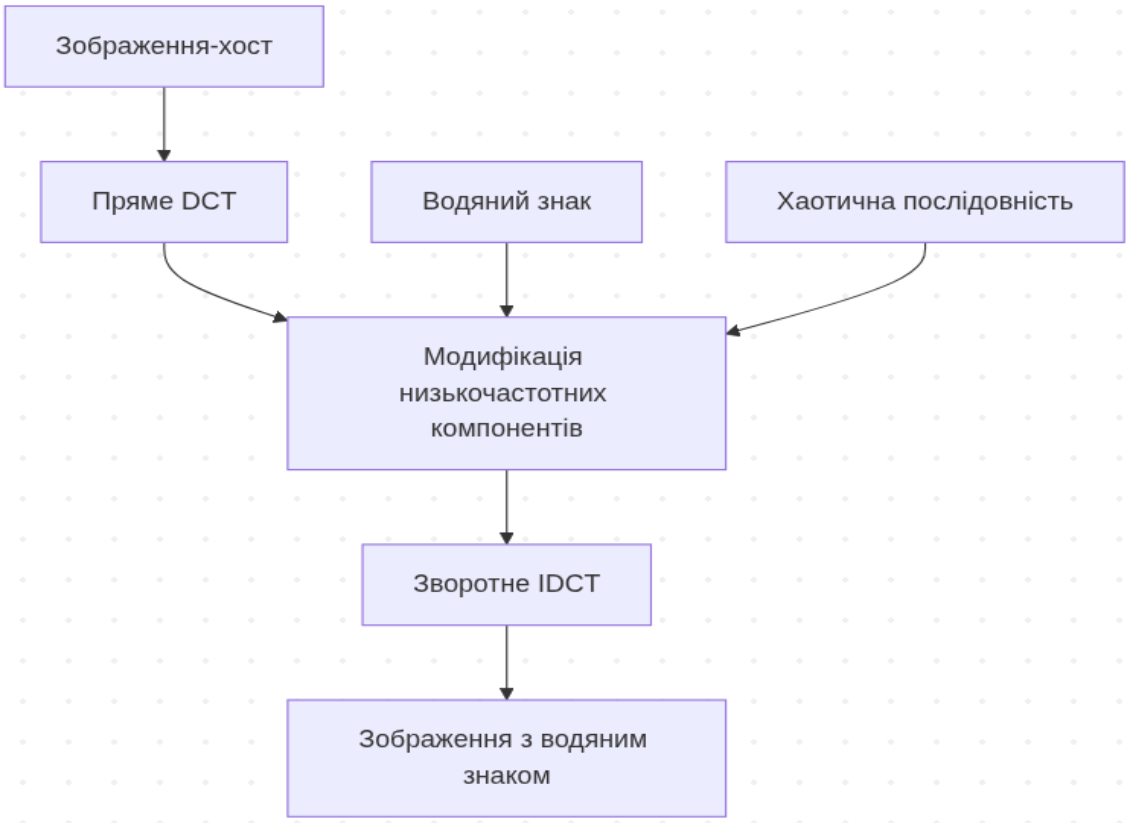

Рисунок 3.4 – Схема роботи функції впровадження водяного знака

Лістинг 3.4 – Впровадження водяного знака

```
import numpy as np
import cv2
def embed_watermark(host_image, watermark, chaotic_sequence,
alpha=0.1):
       # Перевірка розмірів водяного знака та зображення
    if host_image.shape != watermark.shape:
        raise ValueError("Розміри зображення-хоста та водяного
знака повинні збігатися.")
    # Виконання DCT для зображення-хоста
    dct_host = cv2.dct(np.float32(host_image))
    # Модифікація низькочастотних компонентів
    for i in range(watermark.shape[0]):
        for j in range(watermark.shape[1]):
            dct_host[i, j] += alpha * chaotic_sequence[i *
watermark.shape[1] + j] * watermark[i, j]
    # Виконання зворотного DCT (IDCT)
    watermarked_image = cv2.idct(dct_host)
    # Перетворення результату у формат uint8
    return np.uint8(np.clip(watermarked_image, 0, 255))
```

Функція для впровадження водяного знака (лістинг 3.4) приймає кілька вхідних параметрів: host_image – зображення-хост, у яке впроваджується водяний знак, watermark – бінарне зображення водяного знака (значення 0 або 1), chaotic_sequence – хаотична послідовність, яка додає унікальність до процесу впровадження, та alpha – коефіцієнт, що визначає інтенсивність впливу водяного знака на зображення-хост. Ці параметри дозволяють налаштувати процес впровадження відповідно до вимог.

На першому етапі перевіряється, чи збігаються розміри зображення-хоста та водяного знака. Це необхідно для коректної модифікації DCT-коефіцієнтів. Далі зображення-хост перетворюється у частотний домен за допомогою функції cv2.dct, що дозволяє працювати з його частотними компонентами.

На етапі модифікації низькочастотні компоненти DCT-коефіцієнтів змінюються залежно від значень водяного знака (0 або 1) та відповідних значень хаотичної послідовності. Це забезпечує унікальність і стійкість водяного знака.

Після модифікації виконується зворотне перетворення (IDCT) за допомогою функції cv2.idct, щоб отримати зображення з водяним знаком у

просторовому домені. Результат повертається у форматі uint8, що дозволяє зберігати або відображати зображення з водяним знаком.

Для зображення-хоста розміром 256x256 пікселів і водяного знака такого ж розміру функція embed_watermark повертає зображення з водяним знаком, яке візуально майже не відрізняється від оригінального. Водяний знак інтегрується в низькочастотні компоненти, що забезпечує його стійкість до атак.

Ця функція є основою для впровадження водяного знака, забезпечуючи баланс між якістю зображення та стійкістю водяного знака.

3.6 Моделювання атак

Моделювання атак є важливим етапом для перевірки стійкості водяного знака до різних видів обробки зображень. Програма підтримує кілька типів атак, які дозволяють оцінити, наскільки водяний знак зберігається після впливу зовнішніх факторів. Ці атаки включають стиснення JPEG, додавання шуму та фільтрацію.

Кожен із методів моделює реальні сценарії, які можуть виникнути під час обробки зображень, наприклад, при збереженні у форматі JPEG, передачі через мережу або редагуванні.

Основні типи атак (рис. 3.5):

- JPEG-стиснення. Функція simulate_jpeg_compression моделює втрати якості через стиснення у форматі JPEG. Це дозволяє перевірити, чи зберігається водяний знак після зменшення розміру файлу;

- додавання шуму. Функція add_noise додає гаусівський шум до зображення, що моделює вплив випадкових перешкод, наприклад, під час передачі зображення через нестабільний канал зв'язку;

- фільтрація. Функція apply_filter застосовує медіанний фільтр для згладжування зображення. Це моделює обробку, спрямовану на видалення шуму або артефактів.

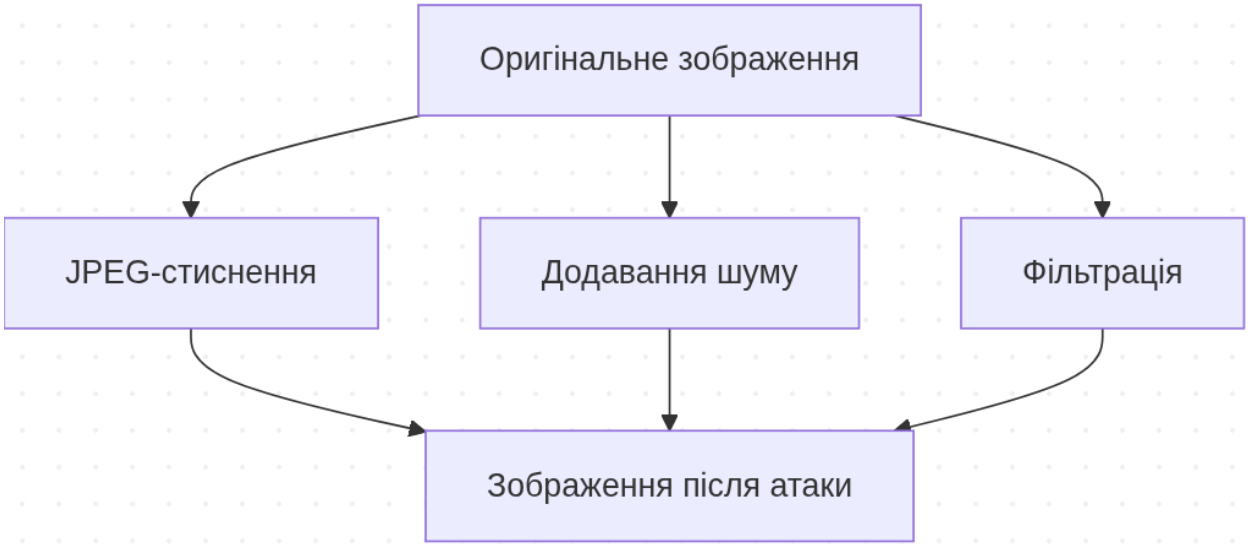

Рисунок 3.5 – Схема роботи блоку моделювання атак

Лістинг 3.5 – Реалізація атак

```
import cv2
import numpy as np
def simulate_jpeg_compression(image, quality=50):
    # Кодування зображення у формат JPEG
    encode_param = [int(cv2.IMWRITE_JPEG_QUALITY), quality]
    _, encoded_image = cv2.imencode('.jpg', image, encode_param)
    # Декодування назад у формат numpy.ndarray
    decoded_image = cv2.imdecode(encoded_image,
cv2.IMREAD_GRAYSCALE)
    return decoded_image
def add_noise(image, mean=0, stddev=10):
   # Генерація шуму
    noise = np.random.normal(mean, stddev,
image.shape).astype(np.float32)
    # Додавання шуму до зображення
    noisy_image = cv2.add(np.float32(image), noise)
    # Обмеження значень у межах [0, 255]
    return np.uint8(np.clip(noisy_image, 0, 255))
def apply_filter(image, kernel_size=3):
      return cv2.medianBlur(image, kernel_size)
```

Функції для моделювання атак реалізують три основні типи впливу на зображення: JPEG-стиснення, додавання шуму та фільтрацію.

JPEG-стиснення виконується за допомогою функції cv2.imencode, яка стискає зображення у формат JPEG із заданим рівнем якості. Після стиснення зображення декодується назад у формат numpy.ndarray за допомогою функції

cv2.imdecode. Це дозволяє моделювати втрати якості, які виникають під час збереження зображення у форматі JPEG.

Додавання шуму реалізується шляхом генерації гаусівського шуму з параметрами mean (середнє значення) та stddev (стандартне відхилення). Згенерований шум додається до зображення, після чого значення пікселів обмежуються в межах [0, 255], щоб уникнути виходу за допустимі границі.

Фільтрація здійснюється за допомогою медіанного фільтра, який застосовується до зображення з використанням заданого розміру ядра kernel_size. Це дозволяє згладити зображення та видалити шум або артефакти.

Результат кожної функції – це зображення після відповідної атаки, яке повертається у форматі numpy.ndarray. Ці функції дозволяють оцінити стійкість водяного знака до різних типів обробки зображень.

Приклад результату:

- JPEG-стиснення – зображення зменшується в розмірі, але втрачає частину деталей;
- додавання шуму – на зображенні з'являються випадкові точки, що імітують перешкоди;
- фільтрація – зображення стає більш згладженим, деталі можуть бути втрачені.

3.7 Тестування

У цьому підрозділі описано основні вимоги до оточення для роботи програми, інструкцію по її запуску, можливі варіанти використання, а також наведено переваги та недоліки реалізованого підходу.

Для коректної роботи програми необхідно забезпечити наступне програмне та апаратне оточення:

Операційна система: Linux, Windows або macOS.

Мова програмування: Python 3.8 або новіша.

Необхідні бібліотеки:

- OpenCV (для обробки зображень): pip install opencv-python;
- NumPy (для роботи з масивами): pip install numpy.

Апаратні вимоги:

- процесор із підтримкою SSE2 (для швидкої обробки зображень);
- мінімум 4 ГБ оперативної пам’яті;
- достатньо місця на диску для збереження проміжних результатів.

Для запуску програми потрібно використати команду:

python main.py --host path_to_host_image.jpg --watermark_text “text”

де:

--host_image – шлях до зображення-хоста.

--watermark – шлях до зображення водяного знака.

Після виконання програми результат буде збережено у вказаному файлі.

Рисунок 3.6 – Стартове зображення

Варіанти для запуску програми:

1 Базовий режим. У цьому режимі програма виконує впровадження водяного знака у зображення-хост із використанням стандартних параметрів:

python main.py –host image.jpg –watermark_text watermark

Маємо такий результат як показано на рисунку 3.7.

```
Водяний знак введено та збережно: watermarked_image.png
Виведення водяного знака виконано: extracted_watermark.png
Метрики якості:
PSNR: 7.58
SSIM: 0.0265
```

Рисунок 3.7 – Результат виконання команди

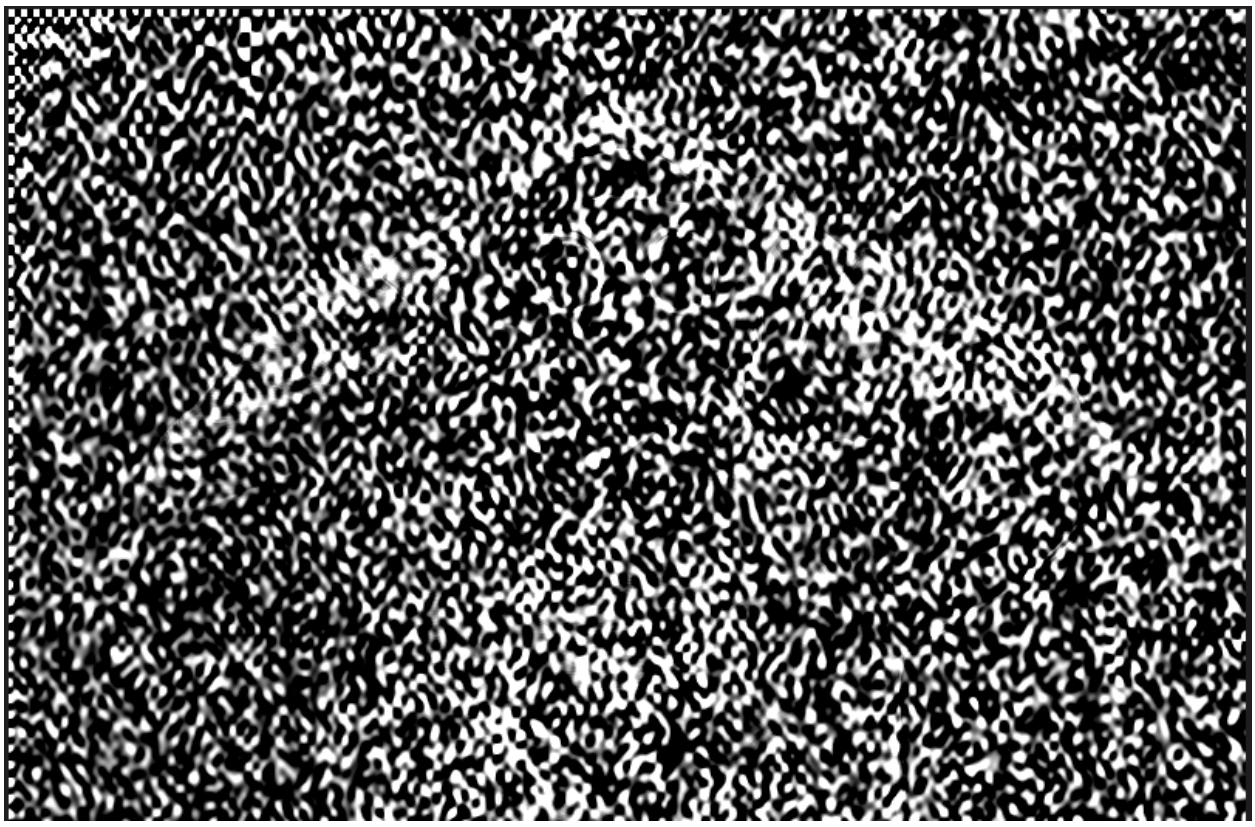

Рисунок 3.8 – Зображення після введення водяного знаку

Результатом стає зображення, на якому висока зашумленість (рис. 3.8).

2 Режим тестування атак. Програма дозволяє змоделювати атаки на зображення з водяним знаком:

- JPEG-стиснення:

python main.py --test_attack jpeg --quality 30

- Додавання шуму:

python main.py --test_attack noise --mean 0 --stddev 20

- Фільтрація:

python main.py --test_attack filter --kernel_size 5

3 Налаштування параметрів впровадження. Можливо змінювати параметри хаотичної карти або інтенсивність впровадження:

python main.py --host image.png --watermark_text "WATERMARK" --alpha 0.1 --wm_width 200 --wm_height 100 --font_scale 1.0 --thickness 2 --seed 0.3 --r 3.99 --attack none

Рисунок 3.9 – Результат виконання програми

Рисунок 3.10 – Варіанти атак на водяний знак

Рисунок 3.11 – Зашумленість зображення

Таблиця 3.1 – Переваги та недоліки

Переваги	Недоліки
Використання хаотичних карт забезпечує високу унікальність водяного знака.	Чутливість до параметрів хаотичної карти (наприклад, значення r та seed).
Модульна структура програми спрощує її розширення та модифікацію.	Залежність якості зображення від розміру водяного знака та параметра alpha.
Підтримка різних типів атак дозволяє оцінити стійкість водяного знака.	Обмежена стійкість до сильних атак, таких як значне стиснення JPEG або агресивна фільтрація.

В таблиці 3.1 показано переваги та недоліки розробленої програми.

ВИСНОВКИ

Внаслідок аналізу застосування хаотичних карт у контексті впровадження цифрових водяних знаків, можна виділити кілька ключових спостережень. Хаотичні карти, що демонструють властивості непередбачуваності та залежності від початкових умов, забезпечують посилений захист від різноманітних впливів, як-от спроби зміни або видалення водяного знака. Завдяки складності відтворення без точних параметрів генерації, такі методи забезпечують більш надійний захист авторських прав та цілісності даних, у порівнянні з традиційними підходами.

Проте, існують й певні виклики, особливо ті, що стосуються чутливості хаотичних карт до змін вхідних даних, а також обмежень у обчислювальних ресурсах. Для підтримки їх стабільності й ефективності, необхідна оптимізація алгоритмів та їх адаптація до різних умов експлуатації.

Використання хаотичних карт у технологіях цифрових водяних знаків демонструє значний потенціал для створення більш складних та захищених методів захисту цифрової інформації. Хаотичні системи, які мають здатність формувати унікальні та комплексні послідовності, можуть значно покращити стійкість до атак та несанкціонованих спроб вилучення водяного знака. Разом з тим, потреба у налаштуванні та оптимізації алгоритмів створює певні труднощі у практичній реалізації цих технологій, проте переваги у порівнянні з традиційними підходами свідчать про їх високу ефективність.

Наступні дослідження в цій сфері можуть бути зосереджені на кількох ключових напрямках. Першим є вдосконалення методів генерації хаотичних карт для досягнення більшої стабільності при мінімальних обчислювальних витратах. Це розширить сферу застосування цих методів у реальних додатках, мінімізуючи вимоги до ресурсів.

Другим важливим напрямом є інтеграція хаотичних карт з іншими сучасними криптографічними методами, такими як цифрові підписи або

методи з відкритими ключами, що дасть змогу розробляти більш надійні та багаторівневі системи захисту. Крім того, використання машинного навчання для адаптації алгоритмів до змін у даних може суттєво поліпшити якість та стійкість водяних знаків.

Методи впровадження водяних знаків на основі хаотичних карт мають великий потенціал у сучасних технологіях забезпечення безпеки. Їх можна використовувати для захисту авторських прав у медіа-індустрії, у фінансових технологіях для забезпечення цілісності транзакцій, а також у галузях, які вимагають високого рівня конфіденційності даних, наприклад, у сфері охорони здоров'я та державної безпеки. З огляду на розвиток квантових обчислень та нових криптографічних стандартів, хаотичні карти можуть стати важливим компонентом для забезпечення довгострокового захисту інформації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Елементи інформатики: довідник / В. С. Височанський, А. І. Кардаш, В. С. Костєв, В. В. Черняхівський. Київ : Наук. думка, 2003. 192 с.

2. Шандиба А.С., Сітніков В. І. Нанесення цифрових водяних знаків з використанням хаотичних карт. Системи управління, навігації та зв’язку . 2024. №4 208-214 с.

3. Anderson R.Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley, 2020, 912 с.

4. Stinson D. Cryptography: Theory and Practice. CRC Press, 2005, 640 с.

5. Preneel B., and Vandewalle J. Watermarking: A Practical Approach. Springer, 2001, 254 с.

6. Cohn S., and Bennett W. Digital Watermarking Techniques and Applications. Wiley, 2019, 432 с.

7. Dufresne G., and Matyjas, D. Introduction to Chaotic Systems and Their Applications in Cryptography. Springer, 2012, 280 с.

8. Cox I. J., and Kilian J. Digital Watermarking and Steganography. Morgan Kaufmann, 2002, 464 с.

9. Li X., and Liu Y. Digital Watermarking: Algorithms and Applications. Springer, 2009, 334 с.

10. O'Connor B., and Hunt T. Chaotic Systems and Cryptography: Applications in Security and Data Integrity. Elsevier, 2016, 250 с.

11. Zhang, M., and Wang, Y. Chaotic Systems in Cryptography. Springer, 2014, 350 с.

12. Schneier, B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. Wiley, 1996, 784 с.

13. Kalker T., and Haitsma, J. Digital Watermarking for Multimedia Applications. Kluwer, 2002, 358 с.