Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет __Комп’ютерної інженерії та управління_____
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки_____
(повна назва)

# КВАЛІФІКАЦІЙНА РОБОТА
# Пояснювальна записка

рівень вищої освіти ____другий (магістерський)_____
(рівень вищої освіти)

Методи комп'ютерної стеганографії для графічних файлів
(тема)

Виконав: студент 2 курсу, групи СКСм-22-1
_____Піскун Я.А._____
(прізвище, ініціали)
Спеціальність 123 Комп’ютерна інженерія_
(код і повна назва спеціальності)

Тип програми __освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)
Освітня програма Спеціалізовані_комп’ютерні системи_____
(повна назва освітньої програми)

Керівник ___доц. каф. АПОТ Рахліс Д.Ю ____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ __Чумаченко С. В._
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проектування обчислювальної техніки

Рівень вищої освіти другий (магістерський)

Спеціальність 123_ Комп'ютерна інженерія
(шифр і назва)

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

Освітня програма Спеціалізовані комп'ютерні системи
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«03» вересня 2023 р.

# ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Піскуну Ярославу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи _ Методи комп'ютерної стеганографії для графічних файлів
затверджена наказом по університету від _03.11.2023 р. № 1282Ст
2. Термін подання студентом роботи до екзаменаційної комісії 22.01. 2024 р.
3. Вихідні дані до роботи
Аналіз стеганоатак та способів захисту від них
Платформи Visual Studio/Webstorm
Мови програмування C#, JS
4. Перелік питань, що потрібно опрацювати в роботі
Стеганографія для графічних файлів
Аналіз існуючих алгоритмів
Створення алгоритму покращення контейнерів до стеганоаналізу
Створення застосунку

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) ___15 слайдів_____

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

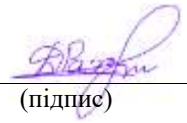
| Найменування розділу | Консультант (посада, прізвище, ім'я, по батькові) | Позначка консультанта про виконання розділу | |
|---|---|---|---|
| | | підпис | дата |
| | | | |
| | | | |
| | | | |

7. Дата видачі завдання _____02.09.2023_____

**КАЛЕНДАРНИЙ ПЛАН**

| № | Назва етапів роботи | Терміни виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Отримання завдання на кваліфікаційну роботу | 02.09.2023-06.09.2023 | *виконано* |
| 2 | Аналіз літератури за темою | 08.09.2023-15.10.2023 | *виконано* |
| 3 | Суть технічної проблеми | 16.10.2023-20.10.2023 | *виконано* |
| 4 | Постановка задачі | 21.10.2023-25.10.2023 | *виконано* |
| 5 | Існуючі методи для вирішення задачі | 26.10.2023-28.10.2023 | *виконано* |
| 6 | Розробка алгоритму застосунку | 29.10.2023-05.11.2023 | *виконано* |
| 7 | Програмна реалізація застосунку | 06.11.2023-07.12.2023 | *виконано* |
| 8 | Тестування застосунку | 08.12.2023-15.12.2023 | *виконано* |
| 9 | Оформлення пояснювальної записки | 16.12.2023-23.12.2023 | *виконано* |
| 10 | Захист кваліфікаційної роботи | 22.01.2024 | |

Студент _____  _____
(підпис)

Керівник роботи  доц. каф. АПОТ Рахліс Д.Ю.
(підпис) (посада, прізвище, ініціали)

# РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи містить 77 сторінок, 13 рисунків, 24 джерела за переліком посилань.

СТЕГАНОГРАФІЯ, ЗАХИСТ ІНФОРМАЦІЇ, СТЕГАНОКОНТЕЙНЕР, ВКРАПЛЕННЯ ІНФОРМАЦІЇ, СТЕГАНОАНАЛІЗ.

Метою роботи є створення покращенного методу стеганографії який буде стійкішим до стеганоаналізу. Було досліджено існуючі популярні методи стеганографії, наведені їх сильні та слабкі сторони. Було вивчено методи стеганоаналізу котрі зараз застосовуються для того щоб визначити чи є інформація у файлах. Визначено потенційне застосування стеганографії та розглянуто питання її створення та становлення у світі.

У роботі розглянуті осноні питання стосовно як проводиться стеганоаналіз, завдяки яким алгоритмам, виявлено проблематику та розроблений алгоритм для покращення стеганоконтейнеру, щоб він був стійкішим до стеганоаналізу. Проведено порівняння між рішеннями які вже існують у нас час та модифікованим алгоритмом.

Розроблений та протестований застосунок котрий дозволяє вкрапувати інформацію до графічних файлів за допомогою створеного алгоритму та зчитувати інформацію зі стеганоконтейнерів. Для написання цього застосунку використана платформа .NET та фреймворк React.

# ABSTRACT

The explanatory note to the qualification work contains 77 pages, 13 figures, 24 sources according to the list of references.

STEGANOGRAPHY, INFORMATION PROTECTION, STEGANOCONTAINER, INFORMATION INJECTION, STEGANOANALYSIS

The purpose of the work is to create an improved method of steganography that will be more resistant to steganoanalysis. The existing popular methods of steganography were investigated, their strengths and weaknesses were given. The methods of steganoanalysis that are currently used to determine whether there is information in files were studied. The potential application of steganography is determined and the issue of its creation and formation in the world is considered.

The work deals with the basic issues related to how steganoanalysis is performed, thanks to which algorithms, problems are identified and an algorithm is developed to improve the steganocontainer so that it is more resistant to steganoanalysis. A comparison was made between the solutions that already exist in our time and the modified algorithm

A developed and tested application that allows inserting information into graphic files using the created algorithm and reading information from stegan containers. The .NET platform and the React framework were used to write this application.

ЗМІСТ

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ

ДКП – дискретне косинусне перетворення.

RS – регулярно-сингулярний (англ. Regular-Singular).

DWT – дискретно хвильове перетворення (англ. Discrete Wavelet Transform).

КС – комп'ютерна стеганографія.

НЗБ – найменш значущий біт.

ШПФ – швидке перетворення Фур’є.

ГМД – гнучкий магнітний диск.

JPEG – об’єднана експертна група з фотографій (англ. Joint Photographic Experts Group).

BMP – бітове зображення (англ. Bitmap Picture)

RLE– кодування довжин серій (англ. Run Length Encoding)

RS – регулярно-сингулярний (англ. Regular-Singular)

# ВСТУП

У сфері інформаційної безпеки, особливо в умовах швидкого розвитку цифрових технологій, з’являється все більше викликів, пов’язаних із захистом конфіденційності даних. Не дивлячись на існування різноманітних криптографічних рішень, багато з яких пропонують високий рівень безпеки, їх складність та вартість часто перешкоджають широкому використанню серед звичайних користувачів. Це створює потребу в розробці більш доступних методів захисту інформації, які б могли бути легко інтегровані в повсякденне життя людей. Методи комп'ютерної стеганографії для графічних файлів стають все більш актуальними в сучасному цифровому світі, де безпека інформації та захист конфіденційності даних відіграють важливу роль. Стеганографія дозволяє приховувати інформацію в графічних файлах таким чином, що візуально відмінності між оригінальним та модифікованим зображенням майже непомітні. Це може мати ряд застосувань, включаючи захист авторських прав, безпечне передавання даних та зберігання конфіденційної інформації [1].

З огляду на те, що більшість сучасних послуг залежать від цифрових мереж та інформаційних систем, ризики, пов’язані з цифровою безпекою, стають все більш актуальними. Несанкціонований доступ, витік інформації, її знищення чи фальсифікація можуть мати серйозні наслідки, включаючи фінансові втрати, порушення приватності та навіть загрози національній безпеці.

Законодавство України визнає ці ризики та визначає кібертероризм, кіберзлочинність, витік державних таємниць, а також маніпуляцію громадською думкою через поширення неправдивої інформації як ключові загрози. Відповідно до цього, існує велика потреба в розробці та впровадженні ефективних та доступних методів захисту інформації, які б могли протистояти

цим загрозам. Стеганографія та стеганоаналіз, який займається приховуванням даних, є одними з таких методів. Вони пропонують альтернативний підхід до традиційної криптографії, дозволяючи передавати конфіденційну інформацію таким чином, що вона залишається непомітною для несанкціонованих осіб. Розвиток цих технологій має велике значення для забезпечення безпеки інформації в умовах сучасного цифрового суспільства.

Метою даного дослідження є проведення огляду існуючих методів комп'ютерної стеганографії для графічних файлів, а також дослідження їх ефективності, стійкості та можливостей покращення. Створення модифікованого алгоритму стеганографії та порівняння його з уже існуючими.

Дослідження в стеганографії має великі перспективи для забезпечення безпеки та конфіденційності інформації в цифровому світі. Розробка ефективних методів стеганографії для графічних файлів може відкрити нові можливості та поліпшити техніки захисту інформації. Дослідження призведе до рекомендацій для вибору та використання найбільш ефективних методів стеганографії, а також розробки власного модифігованого методу. Це допоможе розробляти більш надійні та ефективні системи захисту інформації.

# 1 ОГЛЯД КОМП’ЮТЕРНОЇ СТЕГАНОГРАФІЇ

## 1.1 Становлення поняття “стеганографія”

Стеганографія, давня практика приховування повідомлень, має коріння, що сягають до древнього Єгипту, хоча її принципи можна простежити ще глибше в історії, до часів коли первісні люди залишали послання через наскальні малюнки. Ці ранні форми комунікації можуть бути вважані за прототипи стеганографії, оскільки вони передавали інформацію, яка була прихована від непосвячених.

Літературні джерела першими згадують про стеганографічні методи у творах Геродота, давньогрецького історика. Він описує історію про Демарата, який використовував стеганографію для передачі таємного повідомлення. Демарат видаляв віск з дерев’яних дощечок, писав на деревині, а потім знову покривав їх воском, щоб приховати текст. Цей метод дозволяв передавати інформацію, яка на перший погляд була невидимою.

Інший випадок, який Геродот також згадує, стосується використання раба як носія таємного послання. Рабу голили голову, на шкірі робили татуювання з повідомленням, і коли волосся відростало, він ставав непідозрілим носієм інформації, яку можна було доставити до одержувача без підозри [2].

Стеганографія, як мистецтво приховування інформації, має давню історію, що веде свої корені від древніх цивілізацій до сучасності. У Китаї, наприклад, листи писали на смужках шовку, які потім згортали в кульки, обмазували воском і ковтали кур’єрами, щоб уникнути перехоплення. Цей метод був одним із численних способів, якими люди намагалися зберегти таємниці від сторонніх очей.

У Середньовіччі, часи, які принесли інквізицію та посилене стеження, також стали періодом розвитку стеганографії. Використання шифрів та стеганографічних методів разом стало популярним, а в XV столітті чернець Трітеміус описав багато методів прихованої передачі повідомлень у своїй праці “Steganographia”, яка досі доступна для читання онлайн [3-4].

У XVII-XVIII століттях “чорні кабінети” стали важливими установами, де працювали криптографи, дешифрувальники та хіміки, які використовували невидимі чорнила для перехоплення та розшифровки кореспонденції. Історія монаха Берто, який використав невидимі чорнила для передачі таємного послання, є одним з прикладів інноваційних методів того часу.

Стеганографія також знайшла застосування під час громадянської війни в США, коли агенти передавали інформацію за допомогою спеціального чорнила. Російські революціонери використовували симпатичні чорнила, які згадуються у радянській літературі, а царська охранка зберігала документи з описом використання та перехоплення таємних повідомлень.

Фотографічні мікроточки, які викликали головний біль у спецслужб США під час Другої світової війни, були винайдені значно раніше і вперше застосовані у військових цілях під час франко-пруської війни у 1870 році. Ці мікроточки дозволяли зменшити документи до розміру, який можна було легко приховати та переносити, не викликаючи підозр.

З розвитком технологій стеганографія продовжує еволюціонувати, пропонуючи все більш складні та витончені методи для захисту інформації. Від цифрових водяних знаків до складних алгоритмів, які вбудовують дані в цифрові медіафайли, сучасна стеганографія є важливою частиною кібербезпеки, забезпечуючи конфіденційність у світі, де інформація постійно перебуває під загрозою [5-6].

## 1.2 Особливості стеганографії

Стеганографія, як унікальна форма зв’язку, відрізняється від криптографії тим, що вона приховує не лише зміст повідомлення, але й сам факт його існування [7]. Це означає, що навіть якщо противник підозрює наявність повідомлення, він не може бути впевненим у цьому, оскільки стеганографічні методи дозволяють інтегрувати секретні дані в звичайні повідомлення так, що вони залишаються непомітними.

Слово “стеганографія” походить від грецьких слів “steganos”, що означає “секретний”, та “graphy”, що означає “письмо”. Цей термін охоплює широкий спектр методів таємного спілкування, включаючи використання невидимих чорнил, мікрофотографій, спеціального розташування символів, таємних каналів зв’язку та інших технік. Стеганографія не замінює криптографію, а доповнює її, забезпечуючи додатковий рівень безпеки, оскільки приховування самого повідомлення знижує ймовірність його виявлення [8-10].

З появою комп’ютерів та цифрових технологій стеганографія отримала нове життя. Сучасні стеганографічні методи використовують особливості представлення даних у комп’ютерних файлах та мережах, що дозволяє створювати новітні способи приховування інформації. Комп’ютерна стеганографія, хоча й є відносно новим напрямком, вже має велику кількість публікацій та щорічних конференцій, що свідчить про її активний розвиток.

Модель стеганографічної системи часто описується через “проблему ув’язнених”, яка була запропонована Сіммонсом у 1983 році. У цій моделі двоє учасників намагаються обмінятися секретними повідомленнями, уникаючи втручання охоронця, який контролює канал зв’язку. Ця модель передбачає певні припущення, які можуть полегшувати або ускладнювати обмін секретними даними, залежно від того, чи можуть учасники розділити секретне повідомлення

перед початком спілкування, або чи має охоронець право змінювати повідомлення.

Стеганографія продовжує розвиватися і адаптуватися до сучасних викликів у сфері безпеки інформації. Завдяки новим технологіям, таким як цифрові водяні знаки та складні алгоритми вбудовування даних, стеганографія залишається важливим інструментом у захисті конфіденційності в епоху, коли інформація є надзвичайно цінною та вразливою.

На конференції з приховування інформації у 1996 році було досягнуто консенсусу щодо використання уніфікованої термінології та визначено ключові терміни у цій галузі. Клод Шеннон, відомий своїми роботами у сфері теорії інформації, надав фундаментальні засади для розвитку стеганографії як наукової дисципліни. У контексті сучасної комп'ютерної стеганографії, важливими є два типи файлів: файл-повідомлення, який містить таємну інформацію, та файл-контейнер, у якому це повідомлення може бути приховано. Файл-контейнер може існувати у двох станах: як оригінальний (порожній) файл без прихованої інформації та як результат (заповнений) файл, що містить приховані дані. Ключ у цьому контексті є секретним елементом, який визначає спосіб вбудовування повідомлення у контейнер.

Основні принципи сучасної комп'ютерної стеганографії включають наступне [11-15]:

– методи приховування мають гарантувати автентичність та цілісність файлу, щоб не викликати підозр щодо його змісту;

– припускається, що потенційний противник знає про існування всіх можливих стеганографічних методів, але не має доступу до конкретного ключа;

– безпека стеганографічних методів базується на збереженні основних характеристик відкрито переданого файлу після внесення в нього секретного повідомлення, а також на використанні ключа, який залишається невідомим

противнику;

– якщо противник дізнається про факт приховування повідомлення, вилучення самого секретного повідомлення без знання ключа стає складною задачею, що вимагає значних обчислювальних ресурсів.

Розширюючи ці ідеї, можна сказати, що стеганографія відіграє ключову роль у захисті інформації в епоху цифрових технологій. Вона дозволяє не лише захищати зміст повідомлень, але й приховувати сам факт їх передачі, що є особливо важливим у ситуаціях, коли необхідно уникнути будь-якої підозри щодо існування секретного зв'язку. Завдяки стеганографії, інформація може бути вбудована в зображення, аудіофайли, відео та інші медіа, забезпечуючи додатковий рівень безпеки в сучасному світі, де дані є надзвичайно цінними.

Зі збільшенням впливу всесвітніх комп'ютерних мереж, стеганографія набуває все більшої ваги. Дослідження джерел інформації в мережі Інтернет вказує на те, що сьогодні стеганографічні технології широко застосовуються для таких ключових цілей:

– забезпечення конфіденційності даних та їх захист від неправомірного доступу;

– обходження систем моніторингу та керування мережевими ресурсами;

– маскування програмного забезпечення;

– охорона авторських прав на певні форми інтелектуальної власності.

Забезпечення конфіденційності інформації є одним з основних застосувань криптографії. В якості прикладу, можна розглянути цифрове аудіо: за одну секунду звуку з частотою дискретизації 44100 Гц та 8-бітним стерео можливо приховати близько 10 Кбайт інформації. Це досягається шляхом заміни менш значущих бітів, що призводить до зміни відліків на менше ніж 1%, і така зміна зазвичай залишається непомітною для більшості людей.

Стеганографія також використовується для уникнення моніторингу та управління мережевими ресурсами, що дозволяє обходити контроль над передачею інформації через сервери локальних та глобальних мереж.

Маскування програмного забезпечення є ще однією важливою функцією стеганографії. Це дозволяє приховувати програми під виглядом звичайних програмних продуктів, таких як текстові редактори, або в мультимедійних файлах, наприклад, в аудіодоріжках комп'ютерних ігор, щоб запобігти їх використанню неліцензованими користувачами [16].

Охорона авторських прав на певні форми інтелектуальної власності за допомогою стеганографії полягає у вбудовуванні спеціальних міток у комп'ютерні графічні зображення, які є невидимими для людського ока, але можуть бути ідентифіковані за допомогою спеціалізованого програмного забезпечення. Цей метод використовується для захисту інтелектуальної власності в аудіо, відео та графічних файлах, забезпечуючи захист від несанкціонованого копіювання та розповсюдження [17].

При розробці стеганографічних систем важливо враховувати декілька ключових принципів, розглянутих нижче.

1. Опонент може мати повне розуміння стеганографічної системи та її реалізації, але єдиним елементом, який залишається недоступним для нього, є секретний ключ. Цей ключ дозволяє власнику виявити наявність та зміст прихованого повідомлення.

2. Навіть якщо опонент дізнається про існування прихованого повідомлення, він не зможе отримати доступ до подібних повідомлень у інших контейнерах, доки ключ залишається таємним.

3. Потенційний опонент не повинен мати жодних технічних чи інших переваг, які б дозволили йому виявити або розшифрувати зміст прихованих повідомлень.

Загальна модель стегосистеми показана на рисунку 1.1.

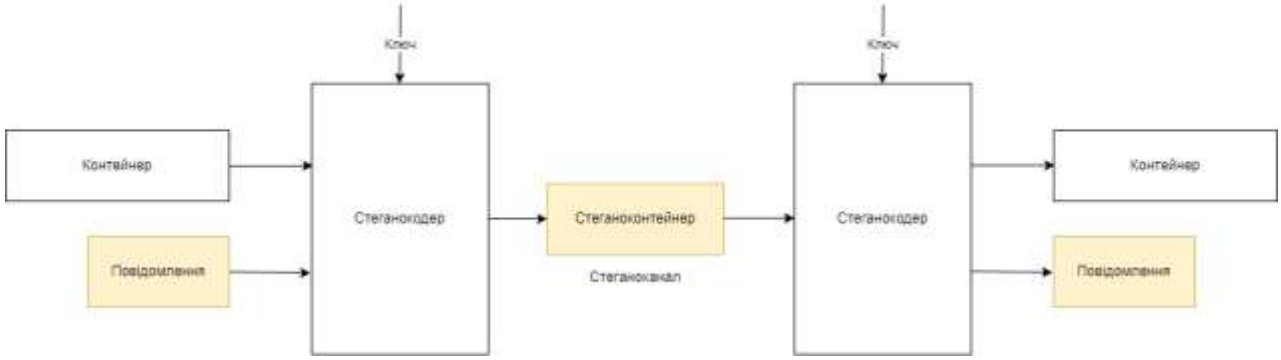



Рисунок 1.1 – Модель стеганосистеми

Для приховування може використовуватися будь-який тип даних, включаючи текст, повідомлення, зображення тощо.

Термін “повідомлення” є універсальним і може означати текст, зображення або аудіодані.

“Контейнер” – визначається як будь-які дані, які використовуються для приховування секретних повідомлень.

Приховане повідомлення – це дані, які інтегруються в контейнер для маскування.

Стеганографічний канал, або стеганоканал, – це метод передачі контейнера, що містить стеганографічні дані.

Стеганоключ, або ключ, – це секретний ключ, який використовується для маскування інформації у контейнері. В залежності від рівня безпеки, який необхідний (наприклад, використання вже зашифрованого повідомлення), стеганосистема може мати один або декілька ключів для забезпечення захисту [18].

Подібно до криптографії, стеганографічні системи можуть бути класифіковані на основі типу використовуваного ключа:

– системи з секретним ключем;

– системи з відкритим ключем.

У стеганографічних системах, які використовують секретний ключ, застосовується один і той же ключ для вбудовування та вилучення прихованих повідомлень. Цей ключ повинен бути визначений до початку обміну даними або переданий через канал, який забезпечує достатній рівень безпеки. Важливо, щоб ключ залишався конфіденційним, оскільки він є основою для забезпечення безпеки передачі інформації.

З іншого боку, стеганосистеми з відкритим ключем використовують два різних ключі: один для вбудовування інформації в контейнер і інший для її вилучення. Ці ключі розроблені таким чином, що неможливо обчислити приватний ключ, використовуючи відкритий ключ. Відкритий ключ можна безпечно передавати через незахищені канали, оскільки він не дає доступу до прихованої інформації без відповідного приватного ключа. Така система забезпечує гнучкість у випадках, коли відправник і отримувач не мають повної довіри один до одного, але все ж потребують безпечного обміну інформацією.

Існує декілька аспектів, вказаних нижче, які треба враховувати при роботі зі стеганосистемою.

1. Модифікація контейнера: властивості контейнера мають бути змінені таким чином, щоб не було можливості визначити зміни при зоровому огляді. Це забезпечує високий рівень приховування вбудованого повідомлення і гарантує, що воно не приверне увагу потенційного нападника під час передачі через комунікаційний канал.

2. Стійкість до спотворень: стеганоповідомлення повинно залишатися незмінним навіть після спотворень, включаючи ті, що є навмисними. Під час передачі, контейнер (наприклад, зображення або звук) може піддаватися різним трансформаціям, таким як зміна розміру або формату, а також стисканню, включаючи стиснення з втратами.

3. Використання коду з виправленням помилок: для забезпечення

цілісності вбудованого повідомлення необхідно застосування кодів, які можуть виправляти помилки.

4. Дублювання повідомлення: для підвищення надійності, вбудоване повідомлення має бути продубльоване, щоб забезпечити його збереження у випадку втрати або пошкодження даних.

Сучасна стеганографія – це наука про приховане зберігання або передачу інформації. Вона включає три основні напрями:

– приховування інформації – це вбудовування секретних даних у звичайний цифровий об'єкт, такий як зображення, звуковий файл або відео. Ці дані можуть бути будь-якого типу, включаючи текст, зображення, звук або код;

– цифрові водяні знаки – це невидимі позначки, які додаються до цифрових об'єктів для захисту їхніх авторських прав або для ідентифікації. Вони можуть бути видимими або невидимими для людського ока;

– ідентифікаційні заголовки – це невеликі фрагменти інформації, які додаються до цифрових об'єктів для їхнього опису або ідентифікації. Вони можуть містити такі дані, як назва файлу, дата створення або розмір.

Приховування інформації часто вимагає великого контейнера, оскільки обсяг вбудовуваної інформації може бути значним. Наприклад, для вбудовування секретного повідомлення в зображення розміром 1 МБ може знадобитися контейнер розміром 10 МБ або більше [18-21].

Цифрові водяні знаки повинні бути надійними та стійкими до спотворень, незважаючи на свій невеликий обсяг. Вони часто використовуються для захисту авторських прав на цифрові твори, такі як музика, фотографії та відео.

Цифрові водяні знаки вимагають складніших методів вбудовування, ніж просте впровадження повідомлень чи заголовків. Це пов'язано з тим, що вони повинні включати ідентифікаційні ознаки файлу, що забезпечують його захист.

Сучасна стеганографія – це наука про приховане зберігання інформації. Вона використовує різні методи для вбудовування секретних даних у звичайні цифрові об'єкти.

Приховування інформації – це найпоширеніший напрям стеганографії. Він використовується для передачі секретних повідомлень або зберігання конфіденційної інформації.

Цифрові водяні знаки використовуються для захисту авторських прав на цифрові твори. Вони можуть бути видимими або невидимими, але завжди стійкі до спотворень. Ідентифікаційні заголовки використовуються для опису або ідентифікації цифрових об'єктів. Вони можуть містити такі дані, як назва файлу, дата створення або розмір [22].

Впроваджувані заголовки – це невеликі фрагменти інформації, які можна додати до цифрових об'єктів без значного погіршення їхньої якості. Вони повинні бути стійкими до основних геометричних перетворень, таких як масштабування, поворот і відображення.

Різні додатки стеганографії вимагають різних балансів між стійкістю вбудованого повідомлення до зовнішніх впливів і його розміром.

Надійність більшості сучасних методів стеганографії збільшується зі збільшенням обсягу вбудованих даних, однак досягає максимуму на певному рівні.

На рисунку 1.2 показана залежність системи від розміру вбудованого повідомлення. Збільшення обсягу вбудованих даних у цифровому контейнері призводить до зниження надійності стеганографічної системи. Це пов'язано з тим, що збільшення обсягу даних вимагає більш значних змін у контейнері, що може призвести до його виявлення стеганоаналітиком.

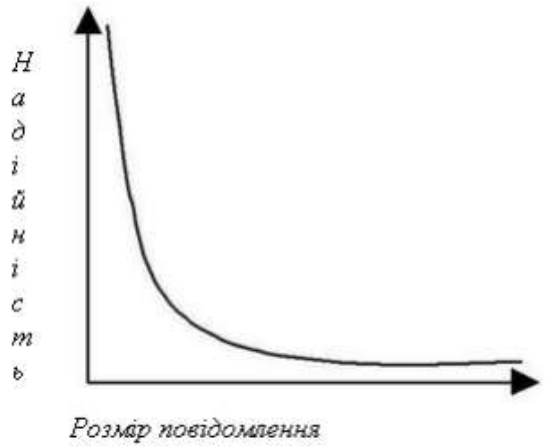



Рисунок 1.2 – Залежність системи від розміру повідомлення

Таким чином, розмір контейнера визначає максимальний обсяг вбудованих даних, при якому система залишається надійною [22].

## 1.3 Стеганоконтейнери

Вибір контейнера для стеганографії має важливе значення для її надійності. Це пов'язано з тим, що різні контейнери по-різному піддаються змінам, викликаним вбудовуванням повідомлення. Наприклад, репродукції відомих картин, як правило, мають характерний вигляд, наприклад, “Мона Ліза” Леонардо Да Вінчі який легко розпізнається досвідченими фахівцями. Це може зробити виявлення прихованої інформації в них більш простим завданням.

Контейнери можна класифікувати за їхньою протяжністю на два типи: безперервні (потокові) та обмежені (фіксованої) довжини. Потокові контейнери не мають чітко визначеного початку або кінця, що ускладнює їхнє виявлення [23]. У потокових контейнерах неможливо передбачити, які будуть наступні бітові значення. Це означає, що приховування даних має відбуватися в реальному часі. Для цього використовуються спеціальні генератори, які визначають відстань між послідовними бітами в потоці.

У потоковому контейнері приховане повідомлення може розташовуватися в будь-якому місці. Це ускладнює його виявлення для одержувача, який повинен знайти початок повідомлення. Якщо контейнер має сигнали або межі, які вказують на початок нового пакета, то таємне повідомлення вставляється безпосередньо за ними. Проте, відправник може мати труднощі, якщо він не знає, чи буде достатньо місця в контейнері для усього прихованого повідомлення.

Коли відправник користується контейнерами певної довжини, він знає, яку кількість даних можна помістити в контейнер. Це дозволяє йому використовувати псевдовипадкову послідовність для вибору прихованих біт. Однак, контейнери фіксованої довжини мають обмежений обсяг, що може призвести до того, що повідомлення не вміститься в контейнер.

Ще одна проблема контейнерів фіксованої довжини полягає в тому, що відстані між прихованими бітами розподіляються рівномірно. Це означає, що більшість біт будуть розташовані близько один до одного, а деякі будуть розділені великими інтервалами. Справжній шум, з іншого боку, має експоненціальний розподіл, що означає, що більшість біт будуть розташовані далеко один від одного, а деякі будуть розташовані близько. Існує спосіб створити псевдовипадкові числа з експоненціальним розподілом, але це може бути складно і трудомістко. Тому контейнери фіксованої довжини часто використовують у практиці, оскільки вони більш прості у використанні. Можливі наступні варіанти контейнерів [24].

1. Контейнери, які генеруються самою стеганосистемою, називаються конструюючими. Прикладом такого контейнера є фрактал Мандельброта, який використовується в програмі MandelSteg. У цьому випадку повідомлення вбудовується в візуальні характеристики фрактала.

2. У селективній стеганографії контейнер обирається з-поміж багатьох

інших контейнерів. Для цього створюється чимало різних контейнерів, а потім відбирається той, що найкраще підходить для схования повідомлення. При обранні найкращого контейнера головною вимогою є натуральність контейнера. Це значить, що контейнер не має виглядати фальшивим або вигаданим. Проте, навіть найкраще підготовлений контейнер дозволяє приховати лише малу кількість даних при дуже великій розмірності самого контейнера. Це є головною проблемою селективної стеганографії.

3. У безальтернативній стеганографії контейнер не вибирається відправником, а надходить ззовні. У цьому випадку відправник не може вплинути на природу контейнера, що може призвести до зниження ефективності стеганографії. Безальтернативна стеганографія часто використовується в ситуаціях, коли відправник не має контролю над контейнером. Наприклад, якщо відправник хоче заховати повідомлення в файлі, який йому не належить.

## 1.4 Мета та завдання кваліфікаційної роботи

Наразі, коли цифрові формати мультимедіа дуже популярні та існують складності з керуванням цифровими ресурсами, дуже важливими стають дослідження в галузі стеганографії [1-6]. Завдання приховання інформації також є актуальною проблемою в умовах розвитку інфраструктури мережевого спілкування користувачів світових комп’ютерних мереж, завдяки яким можна швидко і вигідно надсилати електронні документи в будь-яку точку землі. Однак значна частина переданих матеріалів часто піддається нелегальному копіюванню та поширенню. Тому потрібно шукати способи приховування авторської інформації в різних текстових, графічних, аудіо, відео, та інших видах файлів. Наразі існує багато програмних продуктів, які використовуються для цілей стеганографії і реалізують методи введення таємних даних в різні типи

файлів. Головна мета стеганографії – приховати як зміст, так і сам факт передачі повідомлення. Для цього необхідно, щоб контейнер не змінився занадто сильно після того, як в нього буде вбудовано секретне повідомлення. Це називається прозорістю. Розробники стеганографічних методів повинні знайти спосіб вбудувати секретне повідомлення в контейнер таким чином, щоб не порушити його прозорість. Для цього вони використовують різні методи, такі як зміна кольору пікселів зображення, частоти звуку або інших характеристик цифрового файлу.

Метою цієї кваліфікаційної роботи є глибоке дослідження та аналіз сучасних методів комп'ютерної стеганографії, які використовуються для приховування інформації у графічних файлах.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

– провести аналіз сучасних методів стеганографії для графічних файлів;

– виявити переваги та недоліки кожного методу, а також їх стійкість до атак та виявлення;

– розробити рекомендації щодо вибору методу стеганографії в залежності від, таких як рівень безпеки, обсяг приховуваної інформації, тип графічного файлу та інші фактори;

– запропонувати метод підвищення стеганостійкості;

– визначити ефективність створеного рішення;

– виконати аналіз отриманих результатів.

# 2 МЕТОДИ ТА МОДЕЛІ КОМП'ЮТЕРНОЇ СТЕГАНОГРАФІЇ ДЛЯ ЦИФРОВИХ КОНТЕЙНЕРІВ У ФОРМІ ЗОБРАЖЕНЬ

## 2.1 Огляд та порівняння характеристик стеганографічних методів

Сучасні методи комп'ютерної стеганографії можна розділити на два основні типи [24]:

– методи, засновані на використанні спеціальних властивостей комп'ютерних форматів;

– методи, засновані на надмірності аудіо та візуальної інформації.

### 2.1.1 Методи використання спеціальних властивостей комп'ютерних форматів даних

Поля розширення в мультимедійних форматах зазвичай заповнюються нулями і не використовуються програмами. Перевагою цих методів є простота їх використання. Недоліком цих методів є декілька ступенів скритності та спрооможність передачі тільки невеликої кількості даних

### 2.1.2 Методи спецформатування текстових файлів

Розгляемо методи спецформатування текстових файлів.

1. Використання відомого зміщення слів, речень, абзаців. Ці методи використовуються для зміни порядку розташування рядків і слів у реченні, шляхом вставки додаткових пробілів між словами. Перевагою методу є те, що використання методу є простим, оскільки існує доступне публіковане програмне забезпечення для його реалізації. Недоліками методу є те, що метод має

обмежену продуктивність і може передавати лише невеликі обсяги інформації та ступінь скритності методу є низькою.

2. Методи вибору певних позицій букв (нульовий шифр). Цей метод є окремим випадком методів, заснованих на використанні спеціальних властивостей комп'ютерних форматів. У цьому методі секретне повідомлення кодується за допомогою початкових букв кожного рядка відкритого повідомлення. Перевагою цих методів є простота використання, існує безкоштовне програмне забезпечення, яке можна використовувати для його реалізації. Недоліками цих методів є те, що метод може передавати лише невеликі обсяги інформації та метод має невелику ступінь складності.

### 2.1.3 Методи приховування в невикористовуваних місцях гнучких дисків

Секретна інформація може бути прихована в звичайно невикористовуваних областях ГМД. Перевагою цього методу є простота його використання, завдяки вже існуючим програмним забезпеченням. Недоліками методу є невелика ефективність, невелика ступінь захисту методу, обмежена здатність до захисту інформації.

### 2.1.4 Методи використання імітуючих функцій (mimicfunction)

Цей метод ґрунтується на генерації текстів і є розширенням концепції акровірша. З метою приховання таємного повідомлення генерується осмислений текст, який маскує саме повідомлення. Перевагою цього методу є складність виявлення повідомлення різними перевіряючими системами. Недоліками цього методу є невелика конфіденційність, продуктивність та кількість даних для передачі.

### 2.1.5 Методи видалення ідентифікуючий файл заголовка

Перед тим, як відправити приховане повідомлення, його шифрують за допомогою алгоритму шифрування. Після цього видаляють ідентифікуючий заголовок, який містить інформацію про відправника, отримувача та тип повідомлення. Отримувач заздалегідь інформований про передачу повідомлення і має обмежену кількість заголовків, які він може використовувати для розшифрування повідомлення. Перевагою цього методу простота його використання, завдяки вже існуючим програмним забезпеченням. Недоліком методу є те, що приховування вдається реалізувати частково, тому, що необхідно заздалегідь передати одержувачу певну частину інформації.

### 2.1.6 Методи використання надмірності цифрових фотографії, цифрового звуку і цифрового відео

Заповнення менш значущих бітів цифрових відліків має незначний вплив на сприйняття і практично не впливає на якість. Це надає можливість приховати конфіденційну інформацію. Перевагами цього методу є передача великої кількості інформації та може застосовуватись у різних сферах буденності: захист авторських прав, товарної марки, тощо. Недоліками цього методу є те, що приховування інформації в цифрових потоках призводить до зміни їх статистичних характеристик. Це може бути виявлено за допомогою статистичного аналізу. Щоб зменшити такі виявні ознаки, необхідно виконати корекцію статистичних характеристик цифрових потоків.

Стеганографія має два основних напрямки. Перший напрямок використовує спеціальні властивості комп'ютерних форматів для захисту прихованої інформації від несанкціонованого доступу. Другий напрямок використовує надмірність аудіо та візуальної інформації для приховування інформації в цих даних. Цифрові фотографії, музика та відео представлені

матрицями чисел. Ці числа кодують інтенсивність звуку або зображення в певний момент часу. Однак ці числа не є точними через неточність пристроїв оцифровки та шуми квантування. Молодші розряди цих чисел містять мало корисної інформації, і їх заповнення не впливає на сприйняття. Це дає можливість приховувати додаткову інформацію в цих даних.

Графічні кольорові файли в форматі RGB зберігають кожну точку зображення у вигляді трьох чисел, які представляють колірну інтенсивність червоного, зеленого та синього кольорів. Найменш значущі біти цих чисел містять лише невелику частину інформації про колір точки. Зміна цих бітів незначно впливає на сприйняття зображення. Це дозволяє використовувати ці біти для приховування додаткової інформації в зображенні. Наприклад, можна приховати в зображенні розміром 800 Кбайт близько 100 Кбайт інформації. Ця додаткова інформація буде непомітною при перегляді зображення.

## 2.2 Методи приховування інформації в графічних зображеннях

Методи приховування даних можна розділити на два типи: форматні та неформатні.

Форматні методи використовують особливості формату зберігання даних для приховування інформації. Наприклад, можна використовувати службові поля формату графічних файлів, які не використовуються в поточний момент. Однак такі методи легко виявляються за допомогою автоматичних алгоритмів, тому вони мають низьку стійкість до атак пасивних противників.

Неформатні методи не використовують особливості формату зберігання даних. Вони засновані на зміні даних таким чином, щоб прихована інформація не була помітна. Наприклад, можна змінити значення пікселів зображення або змінити бітовий потік звуку. Неформатні методи мають більшу стійкість до

атак, але вони можуть призвести до погіршення якості даних [27, 28].

### 2.2.1 Неформатні методи приховування в графічних зображеннях

#### 2.2.1.1 Неформатні методи приховування в JPEG

Алгоритм стиснення JPEG перетворює зображення в оптимальний колірний простір, потім субдискретизує його, після чого виконує дискретне косинусне перетворення (ДКП). Квантування коефіцієнтів ДКП дозволяє зменшити розмір файлу, але при цьому може призвести до втрати якості зображення.

Деякі методи приховування даних у файлах JPEG використовують особливості алгоритму стиснення JPEG для того, щоб прихована інформація не була помітна. Наприклад, можна приховати інформацію в вихідних даних зображення, які не піддаються квантуванню.

Один з таких методів полягає у використанні режиму без втрат стиснення JPEG. Цей режим не використовує квантування, тому прихована інформація не впливає на якість зображення.

#### 2.2.1.2 Метод приховування у вихідних даних зображення

У режимі без втрат стиснення JPEG використовується кодування з прогнозуванням. Цей метод полягає в тому, що значення кожного пікселя об'єднується зі значеннями його сусідніх пікселів для формування величини прогнозуючого параметра. Потім отриманий результат віднімається від вихідного значення пікселя.

Це дозволяє зменшити розмір файлу, не втрачаючи якості зображення. Однак воно також створює можливість для приховування даних.

Для цього можна використовувати молодші біти пікселів зображення. Ці біти містять мало корисної інформації, тому їх зміна не впливає на якість

зображення. Таким чином, у режимі без втрат стиснення JPEG можна приховати інформацію безпосередньо у даних самого зображення.

Цей режим майже не застосовується на практиці, тому приховування інформації з використанням цього режиму недоцільне. Це пов'язано з тим, що алгоритм стиснення JPEG з втратами включає в себе ряд етапів, які можуть призвести до втрати прихованої інформації. Наприклад, субдискретизація зменшує кількість біт, які використовуються для кодування кожного пікселя, що може призвести до втрати інформації про колір. Дискретне косинусне перетворення (ДКП) перетворює зображення в просторову частотну область, де прихована інформація може бути легко виявлена. Квантування округлює коефіцієнти ДКП, що також може призвести до втрати інформації.

Інший метод приховування даних у файлах JPEG використовує таблиці квантування. Цей метод є одним з найпоширеніших сьогодні. Ідея полягає в тому, що прихована інформація кодується в молодших бітах чисел, які представляють коефіцієнти квантування.

Перевагою цього методу є те, що він не порушує типову структуру потоку JPEG і є повністю неформатним. Це означає, що його важко виявити за допомогою алгоритмів виявлення стеганографії.

Недоліками цього методу є обмежена кількість таблиць квантування в файлі JPEG (зазвичай одна або дві), що призводить до невеликого обсягу приховуваних даних. Крім того, зміна молодших бітів коефіцієнтів квантування впливає на статистичні характеристики стиснутих блоків, що негативно впливає на ефективність подальшого кодування і призводить до збільшення розміру файлу.

### 2.2.1.3 Метод приховування з використанням таблиць квантування

Цей метод є одним з найпоширеніших методів приховування даних у

файлах JPEG. Він полягає в тому, що прихована інформація кодується в молодших бітах чисел, які представляють коефіцієнти квантування.

Перевагами методу є те, що метод не порушує типову структуру потоку JPEG і є повністю неформатним (це означає, що його важко виявити за допомогою алгоритмів виявлення стеганографії) та те, що метод дозволяє приховувати великі обсяги даних.

Недоліки методу: обмежена кількість таблиць квантування в файлі JPEG (зазвичай одна або дві) та міна молодших бітів коефіцієнтів квантування впливає на статистичні характеристики стиснутих блоків. Це може призвести до збільшення розміру файлу або уповільнення його декомпресії [29].

2.2.1.4 Метод використання неправдивих таблиць квантування

Метод приховування даних у маніпулятивних таблицях квантування є подальшим розвитком попереднього підходу. Він полягає у створенні додаткових таблиць квантування, які містять додаткові відсіки для зберігання прихованої інформації. Це дозволяє значно збільшити обсяг захованих даних.

Однак, цей метод має кілька недоліків. По-перше, він може знизити якість зображення. По-друге, він може бути виявлений алгоритмами виявлення стеганографії. Щоб вирішити ці проблеми, можна використовувати два різновиди цього методу. У першому різновиді створюється кілька додаткових таблиць квантування. Це дозволяє збільшити обсяг захованих даних, але може призвести до подальшого зниження якості зображення. У другому різновиді прихована інформація кодується у вигляді штучних шумів, які додаються до вихідних даних. Це дозволяє збільшити обсяг захованих даних без зниження якості зображення, але може бути виявлено алгоритмами виявлення стеганографії.

Перший різновид методу приховування даних у маніпулятивних таблицях квантування передбачає додавання додаткових таблиць квантування, які дозволяють поліпшити стиснення та зменшити втрати. Цей метод відповідає специфікації алгоритму JPEG, але для більшості зображень використовується невелика кількість додаткових таблиць.

Другий різновид методу полягає в додаванні неправдивих таблиць квантування з певним періодом. Зазвичай використовуються ті ж таблиці, але з варіаціями в найменш значущих бітах, де зберігається прихована інформація. Цей метод також є форматним і не є стійким до пасивних атак, спрямованих на виявлення прихованого повідомлення.

2.2.1.5 Метод приховування в спектрі зображення після квантування

Метод приховування даних у блоках зображення після квантування є одним з найефективніших методів. Він дозволяє приховати значно більшу кількість бітів, ніж попередні методи, і не порушує структуру формату JPEG. Це робить його більш стійким до атак пасивних противників. Метод ґрунтується на тому, що коефіцієнти квантування, які використовуються для стиснення зображення, мають певні статистичні характеристики. Ці характеристики можна використовувати для приховування прихованої інформації. Для цього використовуються класичні методи комп'ютерної стеганографії. Зазвичай прихована інформація кодується у вигляді штучних шумів, які додаються до вихідних даних. Рівень стійкості цього методу залежить від конкретної реалізації. Однак, він, як правило, значно перевищує рівень стійкості попередніх методів.

Кількість прихованої інформації, яку можна зберігати за допомогою цього методу, залежить від розміру стисненого зображення. Чим більше даних приховано, тим більше може бути спотворення зображення. Крім того,

використання великих коефіцієнтів квантування, які необхідні для зберігання більшої кількості даних, також може призвести до спотворення зображення. Зображення, стиснуте за допомогою JPEG, може мати різну якість, залежно від налаштувань. Тому не просто визначити, як впливають на якість зображення такі фактори, як вбудовування даних або використання високих значень квантування. Метод, про який йдеться, має такий принцип. Нехай m – це біти даних, які ми хочемо приховати. Bi, j – це ненульові компоненти блоків квантованого спектра оригінального зображення. Вони розташовані за порядком, який використовується в алгоритмі JPEG, де i – це номер біта компоненти, а j – це номер компоненти. B'i, j – це блоки зміненого зображення. Створюється kj – двійкова послідовність. Якщо наймолодший біт j-го блоку містить наступний біт даних, то kj = 1, а якщо ні, то kj = 0 [23].

Пряме стеганографічне перетворення F: M x B x K → B для даного методу має наступний вигляд:

$$B'_{i_1 j} = \begin{cases} B_{i_1 j}, \forall_i, k_j = 0 \\ m_l,\ i = 0, k_j = 1' \end{cases} \quad (2.1)$$

де $l = \sum_{p=1}^{j} k_p; j = 1,2,3, \dots n$, а відповідне йому зворотне стеганографічне перетворення $F^{-1} : B\ x\ K \rightarrow M$ має вигляд $m_j = B_{0,i}^{l}$,де l таке, що $= \sum_{p=1}^{j} k_p; j = 1,2,3, \dots n$.

2.2.1.6 Методи приховування в графічних зображеннях з палітрою кольорів

За допомогою палітри кольорів можна зробити зображення меншим за розміром. Палітра була корисною в графічних адаптерах, щоб спростити їх будову та підвищити роздільну здатність. Потім палітру почали використовувати в форматах для збереження растрових графічних зображень. Деякі з цих форматів, наприклад GIF, все ще широко використовуються. GIF – це

популярний формат для веб-графіки, бо дозволяє передавати зображення малого розміру. Згодом з’явився формат PNG, який теж може використовувати палітру кольорів, але він поки не став дуже популярним. Таким чином, палітра кольорів в графічних форматах допомагає зменшити розмір зображення, що може бути зручно для передачі даних по мережі або для збереження зображень на пристроях з малою пам’яттю [18].

У звичайних графічних форматах кожна точка зображення містить інформацію про всі три складові кольору (R, G, B). У форматах із палітрою кожна точка зображення містить лише номер кольору з палітри. Палітра кольорів – це набір кольорів, які можна використовувати в зображенні. Таким чином, формати з палітрою дозволяють зменшити розмір файлу, оскільки не потрібно зберігати інформацію про всі кольори зображення.

Нижче можна побачити приклад 8-бітного RGB-зображення розміром 4x4 точки, де кольори точок чергуються між зеленим і синім кольорами в шаховому порядку. Це зображення складається з 16 точок, кожна з яких містить 8 біт інформації про колір. Перша точка містить інформацію про колір зеленого, друга точка містить інформацію про колір синього, і так далі. В результаті, зображення виглядає як шахова дошка, де кожне поле має зелений або синій колір. Це зображення є прикладом того, як можна використовувати палітру кольорів для зменшення розміру файлу зображення.У цьому випадку, палітра містить лише два кольори: зелений і синій. Це означає, що для зберігання інформації про колір кожної точки зображення потрібно лише 1 біт, а не 8 біт, як у випадку з повнокольоровим зображенням. В результаті, це зображення має розмір лише 16 байт, а не 128 байт, як повнокольорове зображення такого ж розміру.

Зображення записано у вигляді матриці з (R, G, B) – елементами:

(255,0,0) (0,255,0) (0,0,255) (0,0,255)
(0,0,255) (0,255,0) (0,0,255) (0,255,0)

(255,0,0) (0,255,0) (0,255,0) (0,0,255)
(255,0,0) (0,255,0) (255,0,0) (0,255,0).

Для збереження наведеної матриці треба 384 біт пам'яті.

Якщо використовувати зображення з палітрою, то для даного зображення потрібна палітра, що складається з двох кольорів:

$$0 \rightarrow (0,255,0);\ 1 \rightarrow (0,0,255).$$

Тоді в цій палітрі зображення набуде вигляду

1 0 0 1
1 0 1 0
1 0 0 1
1 0 0 1

Таким чином, в пам'яті необхідно 48 біт для зберігання інформації про використовувану палітру і 16 біт для зберігання самого зображення.

2.2.1.7 Метод приховування з використанням молодших біт даних зображення

Наведений приклад показує, що метод приховування в молодших бітах не працює для зображень з палітрою, якщо не внести деякі зміни. Це пов'язано з тим, що елементи палітри, номер яких відрізняється лише молодшим бітом, можуть мати абсолютно різні кольори. Якщо замінити молодший біт одного з елементів палітри, то колір цієї точки зображення може різко змінитися. Це призведе до помітних змін самого зображення, що може бути помітно людському оку [24].

Щоб приховати повідомлення в молодших бітах елементів палітри зображення, можна використовувати два основних методи:

– метод з аналізом палітри;

– метод з використанням однакових елементів палітри.

Цей метод використовує аналіз палітри зображення перед тим, як вбудовувати дані. Аналіз палітри допомагає вибрати пари елементів палітри, які

мають невелику різницю в колірних інтенсивностях, не більшу за заданий поріг. Тільки ті точки зображення, які відповідають вибраним елементам палітри, використовуються для вбудовування даних. Перед тим, як витягти дані, знову проводиться аналіз палітри, щоб виявити, які точки зображення були змінені. Але кількість пар елементів палітри, які можна використати для вбудовування, зазвичай дуже мала. Щоб поліпшити цей метод, палітру можна впорядкувати за вагою. Тоді для вбудовування одного біта даних треба змінити усю значимість точки зображення, підставивши нове значення, яке отримано шляхом зміни наймолодшого біта номера впорядкованої палітри.

Цей метод використовує палітру зображення, до якої додаються елементи з однаковими кольорами. Потім він послідовно переглядає всі точки зображення. Якщо точка показує на елемент, який має “близнюка”, вона використовується для вбудовування наступного біта даних. Цей метод не потребує змін у зображенні, але він може збільшити розмір палітри. Обидва методи мають свої плюси та мінуси.

Метод з аналізом палітри дозволяє приховати більше інформації, але він вимагає попереднього аналізу палітри. Метод з використанням однакових елементів палітри не вимагає попереднього аналізу палітри, але він може збільшити розмір палітри.

Вибір методу залежить від конкретних вимог до стеганографічної системи.

Нижче наведено приклад використання цього методу. Нехай повідомлення m = 10010110, палітра складається з двох кольорів: 0 → (0,255,0); 1 → (0,0,255), і зображення має такий вигляд:

0 1 0 1
1 0 1 0
0 1 0 1
1 0 1 0

Після додавання до палітри елемента 2→(0,255,0) у зображенні можна приховати повідомлення m (виділено жирним шрифтом):

1 **2 0** 1
1 1 **0 2**
1 **2** 1 **0**
1 **0 2** 1

В разі додавання декількох однакових кольорів метод можна розширити, але в такому випадку його стійкість зменшиться.

2.2.1.8 Метод приховування шляхом перестановки елементів палітри

Метод приховування шляхом перестановки елементів палітри полягає в тому, що порядок елементів палітри зображення використовується для кодування прихованої інформації. Припустимо, що палітра зображення складається з n різних елементів. Кількість можливих перестановок n елементів становить n!. Це означає, що можна приховати повідомлення довжиною близько log_2(n!) біт. Щоб закодувати повідомлення, потрібно створити відображення, яке за фіксованим ключем встановлює взаємно-однозначне відношення між будь-яким допустимим повідомленням і певною перестановкою елементів палітри [28].

Нижче наведений приклад методу перестановки елементів палітри.
У цьому методі повідомлення *m* представляється цілим числом від 0 до *n! - 1*, де *n* – кількість елементів палітри. Потім елементи палітри переставляються в порядку, визначеному цим числом.

Порядок перестановки визначається наступним чином:

1. Позиція першого елемента палітри визначається як залишок від ділення *m* на *n*.

2. Позиція другого елемента визначається шляхом ділення *m* на *n* без остачі, а потім знаходження залишку від ділення отриманого результату на *n-1*.

3. Таким же чином визначаються позиції інших елементів палітри.

Припустимо, що палітра складається з трьох елементів, впорядкованих в алфавітному порядку: a, b, c. В якості повідомлення ми використовуємо максимально можливе значення m = 3!–1= 5. Залишок від ділення 5 на 3 дорівнює 2, тому в новій палітрі елемент "a" буде займати останню позицію. Далі, [5/3] mod (3–1) = 1 mod 2 = 1. Таким чином, елемент "b" залишається на своєму місці. Очевидно, що елемент "c" повинен зайняти єдине порожнє перше місце. Таким чином, після приховування палітра виглядає як "cba". Вилучення повідомленняДля вилучення повідомлення ми працюємо у зворотному порядку. Позиція першого елемента одиничної палітри "a" займає останнє місце з номером 2, тому залишок від ділення m на 3 дорівнює 2, і m не дорівнює 0. Другий елемент одиничної палітри "b" займає місце з номером 1, тому залишок від ділення m на 2 дорівнює 1. Таким чином, m = 1 + 1 + 2 – 3 = 5.

2.2.1.9 Форматні методи приховування в графічних зображеннях

Форматні методи приховування інформації в графічних файлах використовують особливості формату зберігання цих файлів. Аналізуючи призначення та зміст полів формату, можна знайти такі поля, зміна яких не має помітного впливу на якість зображення. Наприклад, це можуть бути зарезервовані поля, які на даний момент не використовуються або використовуються частково. У такі поля можна розмістити приховану інформацію. Форматні методи є найпростішими в реалізації, однак вони мають істотний недолік: їх легко виявити, якщо відомо справжнє призначення та типовий зміст полів формату. Тому для форматних методів приховування інформації можна розробити повністю автоматизований алгоритм, який буде виявляти факт приховування.

### 2.2.2 Форматні методи приховування в файлах BMP

#### 2.2.2.1 Метод дописування даних в кінець BMP-файлу

Найпростіший метод приховування інформації в графічних файлах формату BMP використовує співвідношення сторін зображення. Стандартні програми для перегляду зображень визначають кінець даних зображення на основі заголовка зображення, який зберігається у підряднику знизу-вгору. Цей метод полягає в тому, що в нижній частині зображення, після даних зображення, можна приховати додаткову інформацію.

Іншим варіантом цього методу є приховування даних після палітри. Палітра зберігається в зображенні перед даними зображення. Початок даних визначається значенням поля "зсув даних". Це поле можна штучно збільшити, а потім використати отриману ділянку BMP-файлу для приховування повідомлення.

У випадках, коли в BMP-файлі зберігається 16-бітне зображення без стиснення, можна скористатися тим фактом, що колірні інтенсивності RGB кодуються за допомогою 5 біт на канал. Це означає, що старший біт кожного 16-бітного значення не містить інформації про колір. Тому його можна використовувати для приховування додаткової інформації.

#### 2.2.2.2 Метод приховування в палітрі

Цей метод приховування інформації в BMP-файлах використовує четвертий байт елементів палітри. Перші три байти кожного елемента палітри використовуються для кодування кольору, а четвертий байт зазвичай не використовується.

Цей метод використовує чотири незайняті байти в заголовку файлу BMP, щоб вбудовувати будь-які дані, не змінюючи розмір файлу. Ці нульові байти не мають значення в форматі BMP, тому їх можна використати для збереження

прихованих даних. Також довжина будь-якої послідовності байтів, що представляє горизонтальний рядок пікселів зображення, повинна ділитися на 4. Це означає, що якщо приховані дані не є кратними 4, їх можна доповнити нульовими байтами до потрібної довжини. Отже, метод вбудовування в нульові байти BMP ґрунтується на тому, що чотири нульові байти в заголовку файлу не використовуються і що довжина будь-якої послідовності байтів, що представляє горизонтальний рядок пікселів зображення, повинна ділитися на 4.

## 2.3 Методи вбудовування інформації в зображення

Стеганографія використовує те, що мультимедійні об'єкти містять інформацію, яка не є необхідною для їх правильного відтворення. Цю надмірність можна використовувати для вбудовування інформації в мультимедійний об'єкт без значного спотворення його зовнішнього вигляду [18].

### 2.3.1 Група методів заміни в просторової області

Методи приховування інформації в зображеннях, що базуються на заміні бітів, називаються просторовими методами.

Найпоширеніший метод просторового приховування інформації - це метод заміни найменш значущих бітів. Цей метод полягає в тому, що біти впроваджуваної інформації замінюють найменш значущі біти пікселів зображення. Оскільки довжина впроваджуваної інформації зазвичай менша за довжину зображення, до впроваджуваної інформації додають випадкові біти, щоб компенсувати різницю.

Перевагами цього методу є його простота реалізації та висока корисна ємність контейнера.

Недоліком методу є те, що будь-які зміни в контейнері, наприклад, компресія, можуть призвести до спотворення впровадженої інформації.

Щоб визначити корисну ємність контейнера при використанні методу заміни найменш значущого біта, можна скористатися формулою, в якій враховуються ширина, висота, кількість компонент кольору та кількість найменш значущих бітів. Іншим просторовим методом є метод випадкового розташування. Цей метод полягає в тому, що біти впроваджуваної інформації розміщують по зображенню з псевдовипадковими інтервалами між ними.

Метод блочного приховування інформації використовує розбиття вихідного зображення на неперекривні блоки заданого розміру. Розмір блоків обирається таким чином, щоб ефективно впровадити секретну інформацію. У кожному блоці обчислюється біт парності.

Додатково, в кожному блоці приховується один біт впроваджуваної інформації таким чином, що якщо біт парності не збігається з секретним бітом, то один з найменш значущих бітів в блоці інвертується, щоб забезпечити збіг біта парності і секретного біта. Перевагами методу блочного приховування інформації є можливість модифікації лише в окремих блоках, можливість вибору розміру блоку, а також можливість мінімізації змін в контейнері. Недоліком методу є його нестійкість до спотворень контейнера.

Стеганографія може використовувати метод заміни палітри для впровадження даних у форматі зображення. Палітра містить список кольорів, які використовуються в зображенні. Кожен колір має індекс, який вказує на його місце в палітрі. Щоб впровадити дані, можна змінити порядок кольорів у палітрі. Наприклад, якщо приховане повідомлення складається з одного біта, його можна впровадити, змінивши порядок кольорів у палітрі так, щоб колір з індексом 0 став першим, а колір з індексом 1 став другим.

Цей метод простий у реалізації, але він має кілька обмежень. По-перше, він підходить тільки для невеликих повідомлень, оскільки кількість можливих послідовностей кольорів у палітрі обмежена. По-друге, цей метод не є стійким до стеганоатак, які включають модифікацію палітри зображення.

Цей метод використовує зміну яскравості для вбудовування даних в просторову область зображення. Для цього зображення-контейнер розбивається на матриці 8x8 пікселів. Кожна матриця поділяється на дві частини B1 і B2. Для кожної частини рахується середня яскравість λ1 і λ2. Вставлення біта даних відбувається за допомогою певної формули, яка залежить від λ1 і λ2.

Цей метод більш стійкий до стеганоатак, ніж метод заміни палітри, оскільки він не вимагає змінювати палітру зображення. Однак він також має обмеження. По-перше, він підходить тільки для невеликих повідомлень, оскільки кількість можливих змін яскравості обмежена. По-друге, цей метод може призвести до погіршення якості зображення. Нижче наведена формула ждя нього

$$S(x,y) = \begin{cases} 1 \text{ при } \lambda 1 - \lambda 2 > \text{E} \\ 0, \text{при } \lambda 1 - \lambda 2 < -\text{E} \end{cases}$$

Параметр E відповідає за порогове значення, яке визначає необхідну різницю між середніми значеннями яскравості.

Метод впровадження біту секретного повідомлення в блок зображення працює так: якщо значення яскравості двох пікселів не відповідають умові, то значення одного з них змінюється так, щоб вони відповідали. Для вилучення впровадженого біту значення яскравості двох пікселів обчислюються. Різниця між ними визначає значення впровадженого біту.

### 2.3.2 Група методів приховування в частотній області

Методи стеганографії, які засновані на заміні, нестійкі до стиснення з втратами, оскільки при стисканні втрачається частина інформації, включаючи і

ту, яка була прихована. Натомість методи приховування в частотній області більш стійкі до стиснення, оскільки вони змінюють не самі дані, а їх частотний розподіл. Ортогональні перетворення, такі як дискретно-косинусне перетворення (ДКП), швидке перетворення Фур'є (ШПФ) і вейвлет-перетворення, часто використовуються в стеганографії, оскільки вони також застосовуються в алгоритмах стиснення зображень. Ці перетворення можуть бути застосовані до всього зображення або лише до його частини.

Існує багато методів стеганографії, які використовують ортогональні перетворення. При виборі методу для впровадження інформації в зображення необхідно враховувати, як воно може бути стиснуто в майбутньому. Наприклад, дискретно-косинусне перетворення (ДКП) є основним алгоритмом стиснення в стандарті JPEG, тоді як вейвлет-перетворення використовується в стандарті JPEG2000 [18].

Метод Коха-Жао, який також називають методом відносної заміни коефіцієнтів ДКП, є одним з найпопулярніших методів стеганографії в частотній області зображення. Він працює шляхом порівняння різниці між двома низькочастотними коефіцієнтами блоків ДКП, на які розбивається зображення. Якщо різниця не відповідає певній умові, то значення одного з коефіцієнтів змінюється так, щоб вона відповідала. Це дозволяє впровадити в блок один біт інформації.

Метод Бенгама-Мемон-Ео-Юнга, який є вдосконаленням методу Коха-Жао, дозволяє впроваджувати інформацію в зображення, не змінюючи всіх блоків. Для цього використовуються лише ті блоки, які мають певні властивості, такі як відсутність різких переходів яскравості та монотонність. Властивість відсутності різких переходів яскравості допомагає уникнути надмірного збільшення значень низькочастотних коефіцієнтів, що може призвести до спотворення зображення. Властивість монотонності означає, що більшість

низькочастотних коефіцієнтів близькі до нуля, що також допомагає зменшити спотворення. Вибір придатних блоків здійснюється порівнянням з параметрами P L і P H. P L обмежує значення першої властивості, щоб значення не перевищувало P L. P H обмежує значення другої властивості, щоб значення не залишалося нижче P H.

Впровадження нуля відбувається, якщо третій коефіцієнт менше будь-якого з перших двох. Впровадження одиниці здійснюється шляхом збільшення третього коефіцієнта відносно перших двох. Якщо такі зміни призводять до значного спотворення блоку, то він не використовується. Цей підхід допомагає знизити спотворення контейнера та впроваджених повідомлень, оскільки він використовується лише для тих блоків, які не спричиняють значного спотворення [21].

Метод Фрідріх – це комбінований метод стеганографії, який використовують два алгоритми впровадження інформації. Один алгоритм використовується для впровадження інформації в низькочастотні коефіцієнти дискретно-косинусного перетворення (ДКП), а інший – для впровадження інформації в середньочастотні коефіцієнти ДКП.

Цей підхід дозволяє досягти високої стійкості до стеганографічних атак, оскільки він робить впроваджену інформацію менш помітною для виявлення.

## 2.4 Реалізація стеганографічних методів

### 2.4.1 Алгоритми стиснення зображень

Цифрові зображення – це складні об'єкти, що складаються з мільйонів маленьких точок, кожна з яких має свій колір. Це робить зображення досить великими файлами. Наприклад, зображення розміром 1024 на 1024 пікселів може займати до 3 мегабайт. Зберігання та передача таких великих файлів може

бути проблематичним. Тому розробка алгоритмів стиснення зображень є важливою задачею. Алгоритми стиснення зображень дозволяють зменшити розмір файлу зображення без суттєвої втрати якості.

Для цього використовують різні методи, наприклад:

– зменшення розміру зображення – цей метод полягає в тому, що зображення зменшується в розмірі, при цьому зберігається його загальна структура.

– використання схожості пікселів – цей метод полягає в тому, що схожі пікселі кодуються одним і тим же значенням.

– використання додаткової інформації про зображення – цей метод полягає в тому, що використовується додаткова інформація про зображення, наприклад, про його яскравість або контраст.

Алгоритми стиснення зображень дозволяють зменшити розмір файлу зображення, зберігаючи при цьому його якість. Існує два типи алгоритмів стиснення зображень: без втрат і з втратами.

Алгоритми без втрат зберігають всю інформацію про зображення, тому відновлене зображення повністю збігається з оригіналом. Вони застосовуються для зберігання зображень, призначених для подальшої обробки, таких як редагування, друк або аналіз.

Алгоритми з втратами дозволяють зменшити розмір файлу зображення за рахунок видалення деякої інформації про зображення. Це може призвести до спотворень, але в деяких випадках вони непомітні для людського ока. Такі алгоритми застосовуються для зберігання зображень, призначених для візуального сприйняття, таких як фотографії або ілюстрації.

Одна з проблем комп'ютерної графіки полягає в тому, що немає єдиного способу оцінити якість стиснення зображення. Для візуального сприйняття важливо, щоб стисле зображення було схожим на оригінал [24].

2.4.1.1 Групове стиснення

Алгоритм RLE (Run Length Encoding) – це простий алгоритм стиснення зображень, який працює шляхом пошуку однакових пікселів в одному рядку. Знайдені ланцюжки однакових пікселів замінюються на пару чисел, що вказують на кількість повторень і значення пікселів. Цей метод може значно зменшити розмір зображення, якщо в ньому є багато повторюваних елементів.

Алгоритм RLE добре працює з зображеннями, в яких є великі області одного кольору, наприклад, з логотипами, схемами або графічними елементами. Однак, в деяких випадках він може призвести до збільшення розміру файлу, наприклад, при збереженні кольорових фотографій, в яких багато різних кольорів.

Групове стиснення зображень – це метод, який дозволяє зменшити розмір зображення, замінюючи послідовності повторюваних значень кольору на пару чисел, що вказують на значення кольору і кількість повторень.

Один з найпоширеніших методів групового стиснення – це PackBits. Він працює так: якщо в послідовності даних зустрічається послідовність однакових значень кольору, то вона замінюється на пару чисел, де перше число – це значення кольору, а друге – кількість повторень.

2.4.1.2 Метод JPEG

Серед методів стиснення зображень з втратами виділяється сімейство JPEG, розроблене експертами з фотографії. Цей метод заснований на частотному аналізі зображення, який дозволяє виділити основні контури і форми зображення, а також дрібні деталі і шум. При стисканні зображення JPEG воно розбивається на невеликі квадратні області розміром 8 на 8 пікселів. Кожна область обробляється незалежно, при цьому видаляються високі частоти, які відповідають дрібним деталям і шуму. Видалення високочастотних компонентів

дозволяє значно зменшити розмір зображення без суттєвої втрати якості. Наприклад, видалення 50% високочастотної інформації призведе до втрати лише 5% корисної інформації, яка міститься в зображенні.

При відновленні зображення JPEG проводиться апроксимація, тобто заповнення відсутніх даних. Однак, оскільки людське око не здатне бачити дрібні деталі, такі втрати не помітні. Метод JPEG широко використовується для зберігання і передачі зображень, таких як фотографії, веб-зображення та відео.

Для стиснення сигналу існує багато різних методів, але найбільш ефективним є дискретне косинусне перетворення (DCT). Цей метод має ряд переваг, включаючи простоту реалізації, високу якість відновлення сигналу і відсутність необхідності роботи з комплексними числами.

DCT працює шляхом розкладання сигналу на ряд частотних складових. Низькі частоти, які відповідають основним елементам сигналу, містяться у верхній частині спектра, а високі частоти, які відповідають дрібним деталям і шуму – у нижній частині. Високі частоти можна обрізати без суттєвої втрати якості, оскільки людське око не здатне бачити їх. Це дозволяє значно зменшити розмір сигналу.

Наприклад можна уявити спектр у вигляді як на рисунку 2.1.

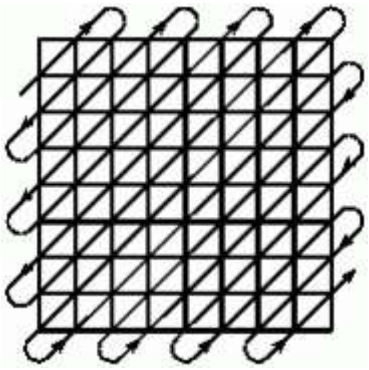


Рисунок 2.1 – Приклад спектру типу «зиг-заг»

Тоді можливо закодувати його за допомогою групового стиснення, а на останьому етапі використати кодування Хафмана.

При використанні алгоритму JPEG застосовується стиснення з втратами, який зберігає не інформацію про колір пікселів, а коефіцієнти розкладання по деякому базису. Коефіцієнти стиснення можуть варіюватися в залежності від якості від 10 до 1000 [28].

Алгоритм JPEG має ряд переваг, включаючи:

– контроль над співвідношенням розмір/якість: користувач може регулювати ступінь стиснення, щоб досягти потрібного балансу між розміром і якістю;

– підтримка глибин кольору до 24 бітів на точку: алгоритм JPEG може обробляти зображення з повною палітрою кольорів;

– великі коефіцієнти стиснення: алгоритм JPEG може значно зменшити розмір зображення без суттєвої втрати якості;

Однак алгоритм JPEG має і ряд недоліків, включаючи:

– можливість початку розпаду зображення за рахунок квантування;

– ефект Гіббса: при квантуванні може виникати ореол навколо різких коольорових переходів;

– алгоритм JPEG є методом стиснення з втратами, що означає, що деяка інформація про зображення втрачається при стисканні. Це може ускладнити або зробити неможливим аналіз або подальшу обробку зображень, оброблених алгоритмом JPEG.

Загалом, алгоритм JPEG є ефективним методом стиснення зображень, який може забезпечити хороший баланс між розміром і якістю. Однак важливо бути обізнаним про його недоліки, щоб приймати зважене рішення про його використання.

Формат JPEG є популярним форматом зображень, який підтримує стиснення. Для того, щоб прихована інформація в зображенні JPEG була стійкою до стиснення, необхідно розробляти алгоритми впровадження, які враховують принципи роботи формату JPEG.

Алгоритм стиснення JPEG працює з колірної моделлю YCbCr, яка відрізняється від адитивної моделі RGB тим, що має три складові: яскравість Y, синього каналу Cb і червоного каналу Cr.

Спочатку, для ефективнішого стиснення, складові зображення в моделі RGB переводяться в модель YCbCr за наступними формулами:

$$\begin{aligned} \mathrm{Y} &= 0.299R + 0.587G + 0.114B, \\ Cb &= -0.169R - 0.332G + 0.5B + 128, \\ Cr &= 0.5R - 0.419G - 0.0813B + 128. \end{aligned}$$

Людське око більш чутливе до зміни яскравості, ніж до зміни кольоровості. Тому для стиснення зображень можна використовувати таку властивість: якщо відкинути інформацію про кольоровість, то можна значно зменшити розмір зображення без суттєвої втрати якості.

Для цього зображення переводять у модель YCbCr, в якій кожен піксель представляється трьома значеннями: яскравістю (Y) та двома компонентами кольоровості (Cb і Cr).

Після цього зображення розбивають на блоки розміром 8 * 8 пікселів. Цей розмір є оптимальним для алгоритму дискретного косинусного перетворення (DCT), який використовується для стиснення зображень JPEG.

DCT перетворює кожний блок зображення в матрицю частотних коефіцієнтів. Ці коефіцієнти містять інформацію про частотний спектр зображення. У матриці DCT низькочастотні коефіцієнти мають більші значення, ніж високочастотні. Високочастотні коефіцієнти відповідають дрібним деталям зображення, які менш помітні для людського ока. Тому для стиснення зображення високочастотні коефіцієнти можуть бути відкинуті без суттєвої

втрати якості. Щоб відкинути високочастотні коефіцієнти, необхідно провести квантування. Квантування – це процес заміни кожного коефіцієнта матриці на дискретне значення. Для квантування використовується матриця квантування, яка є, по суті, матрицею якості. Кожне значення матриці квантування відповідає певному діапазону значень коефіцієнтів матриці DCT.

Матриця квантування заповнюється за наступною формулою:

$$Q(i,j) = 1 + ((1 + i + j) * q),$$

де $q$ – фактор якості, в діапазоні значень [1, 14].

При стисканні зображень JPEG значення фактору якості, задаваного користувачем, визначає ступінь стиснення. Чим більше значення фактору якості, тим більше даних буде відкинуто.

Після дискретного косинусного перетворення (DCT) зображення розкладається на три матриці, які містять інформацію про частотний спектр зображення. Ці матриці називаються матрицями частотних коефіцієнтів.Щоб зменшити розмір зображення, кожну матрицю ділять на матрицю квантування. Матриця квантування визначає кількість значущих цифр, які зберігаються в матриці частотних коефіцієнтів. Елементи отриманих матриць округляються до найближчого цілого числа. Це дозволяє значно зменшити розмір зображення без суттєвої втрати якості. Значущі коефіцієнти матриць частотних коефіцієнтів зосереджені в лівому верхньому кутку. Вони відповідають основним елементам зображення, які сприймаються людським оком. Нулі в правому нижньому кутку матриць відповідають високочастотній інформації, яка не сприймається людським оком. Ця інформація може бути відкинута без суттєвого впливу на якість зображення.

Таким чином, при стиснення зображень JPEG відкидається 51 високочастотна інформація. Квантування має кілька нюансів, які необхідно враховувати при виборі значення фактору якості. Наприклад, при великих

значеннях фактору якості зображення може бути настільки сильно стиснутим, що воно розпадеться на одноколірні блоки. Після квантування в правому нижньому кутку матриці залишаються нулі, які не несуть корисної інформації. Щоб їх відкинути, матриця сканується зиґзаґоподібним способом, починаючи з лівого верхнього кута і закінчуючи в правому нижньому. Отриманий вектор кодується алгоритмом RLE (Run Length Encoding), який дозволяє скоротити розмір даних, заміняючи послідовності однакових символів одним символом, що позначає їх довжину.

Алгоритм RLE (Run Length Encoding) працює так: якщо в векторі є послідовність одного і того ж символу довжиною більше трьох, то вона замінюється на три символи: префікс, який позначає символ, і його кількість. Якщо ж послідовність менша за три, то вона залишається без змін. Ефективність алгоритму залежить від вибору префікса. Найкращим варіантом є вибір префікса таким, який є найрідкіснішим у вхідному векторі. Це пов'язано з тим, що якщо префіксом буде вибраний часто зустрічається символ, то його уявлення буде складатися з трьох символів. Якщо в векторі занадто багато таких символів, то після кодування розмір зашифрованих таким чином даних може виявитися більше розміру вихідних даних.

Інформація, стиснута за допомогою алгоритму RLE, може бути подальше оброблена за допомогою алгоритму Хаффмана. Цей алгоритм призначений для кодування символів таким чином, що менш часто вживані символи отримують довші кодові послідовності. Для кодування можна використовувати як універсальну таблицю кодів, так і спеціалізовану таблицю, розроблену спеціально для даної інформації. При кодуванні даних, отриманих після RLE, важливо враховувати, що великі числа та довгі послідовності нулів зустрічаються нечасто. В результаті роботи алгоритму Хаффмана формується бінарний код, який можна ефективно передавати або зберігати на комп’ютері.

### 2.4.2 Алгоритм методу LSB

Метод LSB (Least Significant Bits – найменш значущий біт) головною темою якого є те, що біти приховуваного повідомлення замінюються на найменш значущі біти зображення. Зміна цих бітів не помітна для людського ока, оскільки вони відповідають за незначні зміни кольору. Принцип роботи методу показаний на рисунку 2.2.

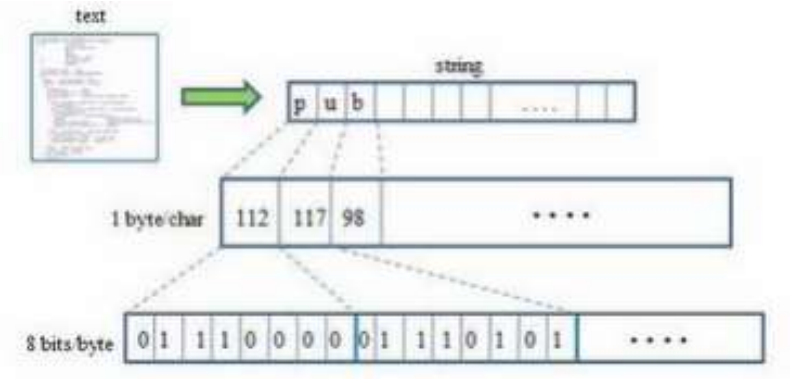



Рисунок 2.2 – Принцип приховування інформації LSB

Зображення у форматі BMP зберігається як матриця значень відтінків кольору для кожної точки зображення. Кожен канал кольору може набувати значень від 0 до 255, що відповідає 24-х бітній глибині кольору. Людське око погано розрізняє незначні зміни кольору. Зміна одного найменш значущого біта в кожному з трьох каналів 24-х бітного кольору призводить до зміни менш ніж на 1% інтенсивності точки, що дозволяє змінювати їх непомітно для ока.

Пропускна здатність методу залежить від кількості бітів, які можна використовувати для завантаження інформації. Якщо використовувати найменш значущі біти в кожному байті, то пропускна здатність становитиме 1/8 розміру контейнера. Якщо використовувати два найменш значущих біти в кожному байті, то пропускна здатність становитиме 1/4 розміру контейнера.

Стеганографічний метод полягає в тому, що приховане повідомлення замінюється на найменш значущі біти пікселів зображення. Зміна цих бітів не помітна для людського ока, оскільки вони відповідають за незначні зміни кольору.

Наприклад, якщо зображення має 24-х бітний формат, то кожен піксель кодується трьома байтами. Кожний байт відповідає одному з каналів кольору RGB. Змінюючи найменш значущий біт в одному з цих байт, ми міняємо значення байта на одиницю.

Такі зміни кольору непомітні для людського ока, оскільки вони складають лише 1/8 від загального значення байта. Крім того, такі зміни можуть взагалі не відобразитися при використанні низькоякісних пристроїв виведення, таких як монітори з низькою роздільною здатністю або принтери з низькою роздільною здатністю.

Метод стеганографії, заснований на LSB, може бути адаптований для використання декількох найменш значущих бітів у кожному байті, що дозволяє збільшити кількість приховуваної інформації. Проте, це також може зменшити рівень прихованості та спростити виявлення стеганографічних змін. Існують варіанти методу LSB, які вирівнюють статистичні відхилення в зображенні, щоб маскувати зміни, внесені під час вбудовування повідомлення.

Інтелектуальні програми для аналізу стеганографії часто перевіряють області з однаковим кольором, тому для підвищення скритності важливо уникати змін у цих пікселях.

Техніки на основі LSB вразливі до атак і працюють ефективно лише в безшумових каналах передачі. Виявлення контейнера з LSB-кодуванням відбувається через аномальні відхилення у розподілі значень найменш значущих бітів цифрового сигналу.

# 3 РОЗРОБКА АЛГОРИТМУ ДЛЯ ПОКРАЩЕННЯ МЕТОДУ ВКРАПЛЕННЯ ІНФОРМАЦІЇ

Стеганографію можна використовувати для передачі конфіденційної інформації по відкритих каналах зв'язку, не привертаючи уваги сторонніх осіб. В інтернеті можна знайти багато безкоштовного програмного забезпечення для стеганографії. Це програмне забезпечення використовує різні алгоритми для вкраплення інформації в контейнери. Непомітність стеганографії може бути використана для злочинних цілей, таких як контрабанда, шпигунство або пропаганда. Стеганоаналіз – це метод виявлення прихованої інформації в контейнерах. Основна задача стеганоаналізу – встановити факт існування прихованої інформації. Стеганоаналіз може використовуватися для захисту від несанкціонованого використання стеганографії.

Виявлення прихованої інформації в контейнерах, які були зашифровані за допомогою різних методів стеганографії, є складним завданням. Деякі методи стеганографії, такі як метод послідовної заміни найменш значущих біт, легко виявляються за допомогою стандартних методів стеганоаналізу, таких як метод на основі критерію хі-квадрат. Однак інші методи стеганографії, такі як метод псевдовипадкового вибору молодших біт, більш важко виявити. Щоб підвищити ефективність стеганоаналізу, необхідно використовувати комплекс методів, які дозволяють виявити різні типи прихованої інформації. Ось кілька прикладів методів, які можна використовувати для виявлення прихованої інформації:

– методи на основі статистичних характеристик контейнера – ці методи використовують знання про статистичні характеристики контейнерів, таких як розподіл бітових значень або розподіл кольорів;

– методи на основі штучного інтелекту, вони використовують алгоритми машинного навчання для виявлення аномалій у контейнерах, які можуть вказувати на наявність прихованої інформації;

– методи на основі експертних знань – використовують знання експертів про стеганографію для виявлення прихованої інформації.

Вибір методів стеганоаналізу залежить від типу прихованої інформації, яку необхідно виявити, а також від можливостей і обмежень конкретного стеганоаналітичного інструменту.

Як відомо, чим більше інформації вкраплюється в контейнер, тим більш ймовірно, що вона буде виявлена. Це пов'язано з тим, що вкраплення інформації може призвести до зміни статистичних характеристик контейнера. Якщо ці зміни є значними, їх можна виявити за допомогою стеганоаналізу. Таким чином, проблема полягає в тому, щоб вкрапити якомога більше інформації в контейнер, не змінюючи при цьому його статистичні характеристики.

Ось кілька способів вирішення цієї проблеми:

– використовувати методи стеганографії, які є менш чутливими до змін статистичних характеристик контейнера.

– використовувати методи стеганографії, які використовують дані контейнера для маскування змін, внесених вкрапленням інформації.

– використовувати методи стеганографії, які використовують штучний інтелект для вибору оптимальних місць для вкраплення інформації.

Важливо знайти баланс між обсягом прихованої інформації та надійністю її передачі. Якщо вкраплення інформації буде занадто великим, його буде легко виявити. Якщо вкраплення інформації буде занадто маленьким, його буде важко відновити в разі перехоплення контейнера.

## 3.1 Математична модель стеганосистеми

Процес звичайного стеганографівчного перетворення описується:

$$E: C \times M \rightarrow S; \tag{3.1}$$

$$D: S \rightarrow M, \tag{3.2}$$

де S – множина контейнерів, D, C, M – окремі контейнери

Залежність (3.1) описує, як приховати інформацію в контейнері, а залежність (3.2) – як витягти приховану інформацію з контейнера. Одна з умов, яка повинна виконуватися для того, щоб ці дві залежності працювали, полягає в тому, що вони не повинні перетинатися. Це означає, що інформація, яка використовується для приховування, не повинна бути такою ж, як інформація, яка використовується для витягування. Наприклад, якщо для приховування використовується метод LSB, то для витягування не можна використовувати той же метод. Якщо ці дві залежності будуть перетинатися, то це може призвести до того, що прихована інформація буде загублена або пошкоджена. Стеганографічна система – це сукупність компонентів, які дозволяють приховати інформацію в інших файлах, таких як зображення, аудіо або відео. Контейнери обираються таким чином, щоб зміни, внесені при вкрапленні інформації, були непомітні. Стеганографічна система вважається надійною, коли прихована інформація не може бути виявлена за допомогою стеганоаналізу. Контейнери можна обирати довільно або підбирати найбільш придатний контейнер для конкретного випадку.

Пряме та зворотне перетворення повинні бути взаємно оберненими і виконуватися таким чином, щоб незначні зміни контейнера не приводили до втрати або спотворення прихованої інформації. Це означає, що якщо в контейнер внести невелике викривлення, то прихована інформація повинна залишитися незмінною.

## 3.2 Метод найменш значущого біту

Стеганографія дозволяє здійснювати таємне спілкування шляхом вбудовування даних у інші медіафайли, такі як зображення, аудіофайли, відео та інші типи файлів. Вона включає в себе три ключові компоненти: контейнер (файл, що маскує повідомлення), секретне повідомлення (інформація для приховування) та стеганоконтейнер (результат вбудовування повідомлення в контейнер). В стеганографії дані часто приховуються у кольорових бітах пікселів у зображенні, де кожен піксель містить червоний, зелений та синій компоненти, кожен з яких може мати значення від 0 до 255 і представлений 24 бітами.

Метод вставки найменш значущого біта (LSB) є одним із способів приховування інформації, де кожен піксель може містити до 3 бітів прихованої інформації, оскільки кожен з трьох кольорових компонентів має 8 бітів. Цей метод полягає у зміні LSB кожного кольорового компонента відповідно до бітів секретного повідомлення. Наприклад, якщо біти повідомлення є 01001010, вони можуть бути приховані шляхом зміни LSB у пікселях зображення. Цей метод є дискретним, оскільки внесені зміни є мінімальними і важко помітними, але він має обмеження у кількості інформації, яку можна зберегти, оскільки кожен піксель може містити лише обмежену кількість бітів.

Наприклад, сітка для 24-бітового зображення може виглядатиме так:

10101101 10011000 11010010 (173, 152, 210)

10110000 11010110 11001010 (176, 214, 200)

10100100 10011000 11000100 (180, 152, 196).

Повідомленням буде ‘Y’: ASCII значення ‘Y’ 89 = 01011001. Після застосування методу стеганоконтейнер виглядатиме так:

1010110**0** 1001100**1** 11010010 (172, 153, 210)

10110000 11010111 11001010 (177, 215, 200)

10100100 10011001 11000100 (180, 153, 196).

Як видно з результатів, зміни у контейнері ледь помітні, тому інформацію складно знайти на перший погляд.

## 3.3 Розподіл секрету Шаміра

Схема Шаміра заснована на тому, що для відновлення секрету, який розділений на кілька часток, потрібно мати не менше, ніж k часток. Кількість часток k визначається ступенем многочлена, який використовується для кодування секрету. Якщо відомо менше, ніж k часток, то інтерполяція многочлена буде неможливою, і секрет не можна буде відновити.

Алгоритм можна умовно розділити на 3 етапи.

1. Підготовчий етап. Обирається випадкові коефіцієнти $S_1, S_2, S_3, S_4, \dots, S_{k-1} \in Z_p$ та складається многочлен (3.3):

$$S(x) = S_{k-1}X^{k-1} + S_{k-2}X^{k-2} + \cdots + S_1X + M \bmod p \quad (3.3)$$

де M – розділяючий секрет, коефіцієнти – довільні елементи. Далі обираються n ненульових несекретних елементів $r_1, r_2, \dots, r_n$ із $Z_p$

2. Розподіл секрету. Обчислюється наступний многочлен (3.4):

$$S(x) = S_{k-1}X^{k-1} + S_{k-2}X^{k-2} + \cdots + S_1X + M \bmod p \quad (3.4)$$

3. Відновлення секрету. Для цього потрібно застосувати інтерполяційну формулу Лагранжа. Інтерполяційна формула Лагранжа дозволяє побудувати многочлен, який проходить через задані точки. У випадку схеми Шаміра, учасники схеми мають k часток секрету. Кожна частка – це точка, в якій проходить секретний многочлен. Щоб відновити секрет, учасники схеми об'єднують свої частки і використовують їх для інтерполяції секретного многочлена формула (3.5).

$$S(x) = \sum_{j=0}^{k-1} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j} \quad (3.5)$$

У схемі розподілу секрету секретний многочлен має таку властивість, що його значення в точці x=0 дорівнює секрету M (3.6):

$$M = \sum_{i=0}^{k-1} c_i S_i, \text{де } S = \prod_{j \neq i} \frac{r_j}{r_j - r_i} \quad (3.6)$$

З описаного у формулах 3.5 та 3.6 стає зрозуміло, що чим більше значення порога, тим повільнішим стає обчислення.

## 3.4 Метод стеганоаналізу “Хі квадрат”

Метод розпізнавання формату зображення аналізує частоту появи пар значень в наборі даних елементів зображення. Для BMP файлів парами значень є значення пікселів зображення, для JPEG – квантовані коефіцієнти дискретного косинусного перетворення, які відрізняються за молодшим бітом. Цей метод є ефективним, оскільки він є незалежним від розміру і розширення зображення.

Молодші біти зображень не є рівномірно розподіленими. Частоти двох сусідніх елементів зображення мають бути досить різними. Коли інформація вбудовується у файл, частоти окремих бітів можуть наближатися одна до одної або навіть ставати однаковими. Атака хі-квадрат ($\chi^2$) використовує цей статистичний феномен для ідентифікації прихованих даних. Цей метод аналізує частоту зустрічання парних та непарних бітів у файлі. Якщо виявлено, що ці частоти є схожими, це може свідчити про наявність вбудованої інформації. Атака хі-квадрат є ефективною для різних типів стеганографічного програмного забезпечення, але вона не є бездоганною і може бути неефективною проти деяких більш складних методів стеганографії.

Як видно з рисунка 3.1, при послідовній заміні НЗБ елементів, метод легко знаходить повідомлення, а при псевдовипадковому не працює, що можна побачити на рисунку 3.2

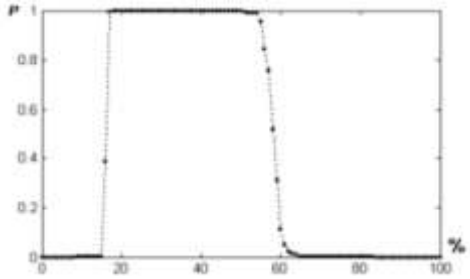



Рисунок 3.1 – Ймовірність повідомлення при послідовній заміні НЗБ

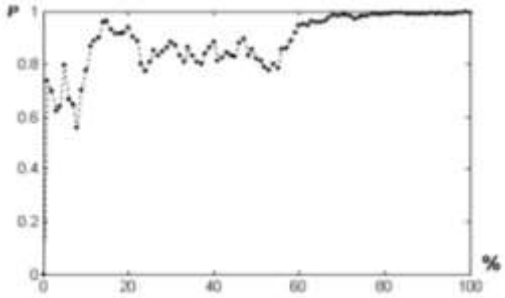



Рисунок 3.2 – Ймовірність повідомлення при псевдовипадковій заміні НЗБ

## 3.5 Метод стеганоаналізу RS-атака

Метод полягає в тому, що зображення розбивається на групи пікселів. Для кожної групи визначається функція, яка характеризує її однорідність. Якщо в зображення вбудовано стеганографічне повідомлення, то функції регулярності для груп пікселів, що містять стегобіти, будуть відрізнятися від функцій регулярності для груп пікселів, що їх не містять.

RS метод ґрунтується на тому, що в природному зображенні розподіл біт у групі не змінюється, якщо значення пікселів у групі зсунути на одиницю. Якщо в зображення вбудовано стеганографічне повідомлення, то розподіл біт у групі,

що містить стегобіти, буде відрізнятися від розподілу біт у групі, що не містить стегобіти. Цей факт можна використати для виявлення стеганографії.

Розглянемо, що відбувається з молодшими бітами зображення, якщо ми повністю переписуємо їх бітами повідомлення. Якщо ми вбудуємо випадкове повідомлення довжиною, рівною розміру зображення, то 50% молодших біт будуть інвертовані. На рисунку 3.3 представлена діаграма, яка показує, як змінюється відносне співвідношення регулярних і сингулярних груп зображення в залежності від кількості інвертованих біт.

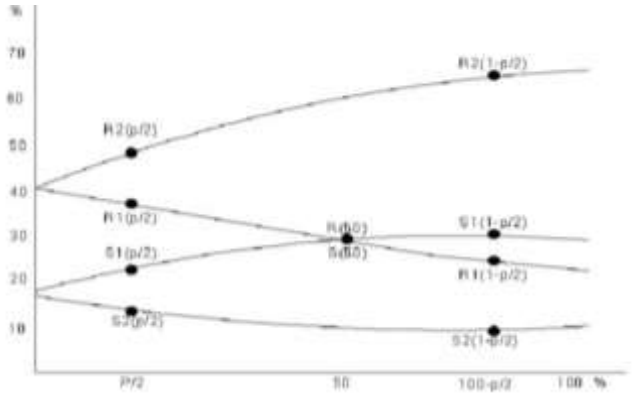


Рисунок 3.3 – Залежність відношення груп від кількості інвертованих бітів

Якщо в зображення вбудовано повідомлення довжиною p біт, при цьому 50% молодших біт будуть інвертовані, то на діаграмі це відповідатиме точці p/2.

## 3.6 Опис розробленого алгоритму

Нижче наведений покроковий опис розробленого алгоритму, який буде краще захищати від наведених стеганоатак:

– підготувати всі дані: ключ, повідомлення яке буде вкраплено, стеганоконтейрер та визначити кількість сторін розподілу;

– застосувати схему розподілу Шаміра до початкового повідомлення;

– використовувати генератор псевдовипадкових чисел для контролю вкраплення інформації у контейнер, вкрапити усі розподіли у копію початкового контейнера та передати контейнер учасникам;

– отримати стегоконтейнери з розподілами;

– відновити повідомлення вибравши будь-яку кількість стегоконтейнерів, що не менше за кількість сторін, необхідних для відновлення розподілу та використати розподіли зі стегоконтейнерів для відновлення початкового повідомлення.

Графічна схема роозробленого алгоритму представлена на рисунку 3.4.


Рисунок 3.4 – Графічна схема створеного алгоритму.

# 4 РЕАЛІЗАЦІЯ АЛГОРИТМУ ТА ОПИС РОЗРОБЛЕНОГО ЗАСТОСУНКУ

## 4.1 Засоби розробки серверної частини

Розробка серверної частини була за зроблена за допомогою платформи .NET, та мови програмування C#. Вибір .NET як основи для серверної частини застосунку не є випадковим, а є результатом ретельного аналізу його переваг. Цей фреймворк відомий своєю високою продуктивністю, що є вирішальним фактором для розробки надійних і ефективних серверних додатків. Його надійність забезпечується стабільною роботою в різних умовах, що дозволяє підтримувати безперебійну роботу критично важливих систем.

Крос-платформенність .NET є однією з його ключових характеристик, яка дозволяє розробникам створювати додатки, що працюють на різних операційних системах без необхідності зміни коду. Це нетільки спрощує процес розробки, але й забезпечує легкість управління та розгортання додатків, а також дає можливість вибору оптимальної операційної системи для конкретних потреб.

Екосистема .NET, що включає в себе велику кількість бібліотек та інструментів, сприяє швидкому розвитку додатків. Розробники мають доступ до готових рішень для різноманітних завдань, що значно скорочує час на розробку та тестування. Підтримка спільноти та постійні оновлення забезпечують актуальність технологій та відповідність сучасним вимогам.

Безпека даних та додатків є однією з найважливіших вимог сучасного програмного забезпечення, і .NET пропонує розширені можливості для її забезпечення. Вбудовані механізми безпеки дозволяють захистити інформацію від несанкціонованого доступу та забезпечити конфіденційність обробки даних.

Масштабованість .NET дозволяє легко адаптувати додатки під зростаючі навантаження, що є важливим для бізнесів, які планують розширення та збільшення клієнтської бази. Фреймворк оптимізований для роботи з великими обсягами даних та високою кількістю запитів, що робить його ідеальним вибором для розробки масштабованих серверних рішень.

Таким чином, .NET є вибором, що обумовлений потребами високої продуктивності, надійності, безпеки, гнучкості та масштабованості, що робить його ідеальною платформою для розробки серверної частини застосунків. Ці характеристики, разом із потужною підтримкою спільноти та постійними оновленнями, забезпечують високу якість та довгострокову перспективу розвитку проектів на базі .NET.

C# було обрано як головну мову програмування для даного проекту завдяки наявності розширених наукових та графічних бібліотек. Відмінною альтернативою є лише Java. C# підтримується на багатьох платформах, включаючи Windows, UNIX та Macintosh, і працює на них однаково добре. Як скриптова мова, вона сприяє швидкому прототипуванню та вирішенню проблем, що є корисним для наукових досліджень та розробки. Синтезуючи найкращі характеристики сучасних мов програмування, таких як Java, C++ та VisualBasic, C# стає не просто агрегатом їхніх переваг, а мовою нової ери.

## 4.2 Засоби розробки клієнтської частини

Вибір React для розробки клієнтської частини застосунку є свідомим рішенням, що базується на його унікальних характеристиках та перевагах. React, бібліотека для створення користувацьких інтерфейсів, розроблена Facebook, відзначається своєю декларативністю, ефективністю та гнучкістю. Використання React дозволяє створювати великі веб-додатки, які можуть

оновлювати дані без перезавантаження сторінки, забезпечуючи швидку відповідь інтерфейсу на дії користувача.

Однією з ключових особливостей React є віртуальний DOM, який оптимізує процес оновлення інтерфейсу, зменшуючи кількість взаємодій з реальним DOM та підвищуючи продуктивність додатку. Компонентний підхід React сприяє підвищенню перевикористання коду та спрощує процес розробки, дозволяючи розробникам створювати ізольовані частини інтерфейсу, які можна легко інтегрувати та тестувати.

React також підтримує сучасний підхід до розробки через використання JSX, синтаксису, який дозволяє писати HTML-структури в JavaScript-коді, роблячи код більш читабельним та легким для розуміння. Це, разом із потужною екосистемою інструментів, таких як Redux для управління станом, робить React вибором, який забезпечує швидку розробку, високу продуктивність та легкість підтримки великих та складних веб-додатків.

Таким чином, вибір React для клієнтської частини застосунку обумовлений його здатністю забезпечити ефективну взаємодію з користувачем, високу продуктивність, гнучкість у розробці та легкість інтеграції з іншими системами та бібліотеками, що робить його ідеальним інструментом для створення сучасних динамічних веб-додатків.

## 4.3 Опис розробки web-застосунку

Для підвищення безпеки інформації, вбудованої методом LSB, можна використовувати криптографічні методи, наприклад, шифрування повідомлення перед його впровадженням. Також можна застосувати схему розподілу секрету Шаміра, щоб отримати кілька ключів, необхідних для розшифровки повідомлення.

В даній роботі застосована модифікація методу LSB, яка використовує функцію, що генерує випадкові значення порядкових номерів пікселів. Це дозволяє впроваджувати інформацію в зображення у випадковому порядку, що ускладнює її виявлення і розшифровку.

Після завантаження зображення визначається кількість біт, які можна використовувати для вбудовування інформації. Введений користувачем текст представляється в бінарному вигляді, де кожний символ кодується різним числом біт залежно від його алфавіту. Наприклад, літери російського алфавіту кодуються двома байтами, а букви латинського алфавіту, арабські цифри і розділові знаки - одним байтом.

Після встановлення максимальної кількості бітів, які можуть бути використані для внесення інформації, зображення ділиться на окремі кольорові канали – червоний (R), зелений (G) та синій (B). Далі створюється псевдовипадкова послідовність для визначення порядку пікселів, які будуть використовуватися для вставки даних. Біти секретного повідомлення вставляються в пікселі згідно з цією послідовністю, змінюючи тільки один найменш значущий біт у кожному кольоровому компоненті. Якщо потрібно вставити нуль і відповідний біт вже є нулем, або потрібно вставити одиницю і біт вже є одиницею, то кольоровий компонент залишається без змін. Цей процес враховується при обчисленні співвідношення сигнал-шум, яке базується на кількості модифікованих пікселів. Зображення з вбудованою інформацією формується шляхом копіювання оригінального файлу та внесення в нього змін. Початкове зображення залишається без змін, а модифіковане зберігається окремо. Для вилучення інформації зі стеганоконтейнера процес відбувається у зворотному порядку: спочатку визначаються кольорові компоненти всіх пікселів, потім розраховується довжина повідомлення, закодована на початку, і врешті-решт вилучається повне повідомлення.

## 4.4 Опис роботи застосунку

Для запуску веб застосунку треба перейти на його сторінку у браузері. Після цього відкриється форма на якій можна побачити головну форму для користування застосунком.

Зліва знаходиться форма для вибору зображень для того, щоб вкрапити в нього повідомлення або обрати декілька файлів, щоб його отримати.

Вкладка “Encode” відкривається за замовченням (рис. 4.1), тут можна ввести бажане повідомлення для того, щоб його вкрапити, та налаштування кількості сторін для розподілу для схеми Шаміра.

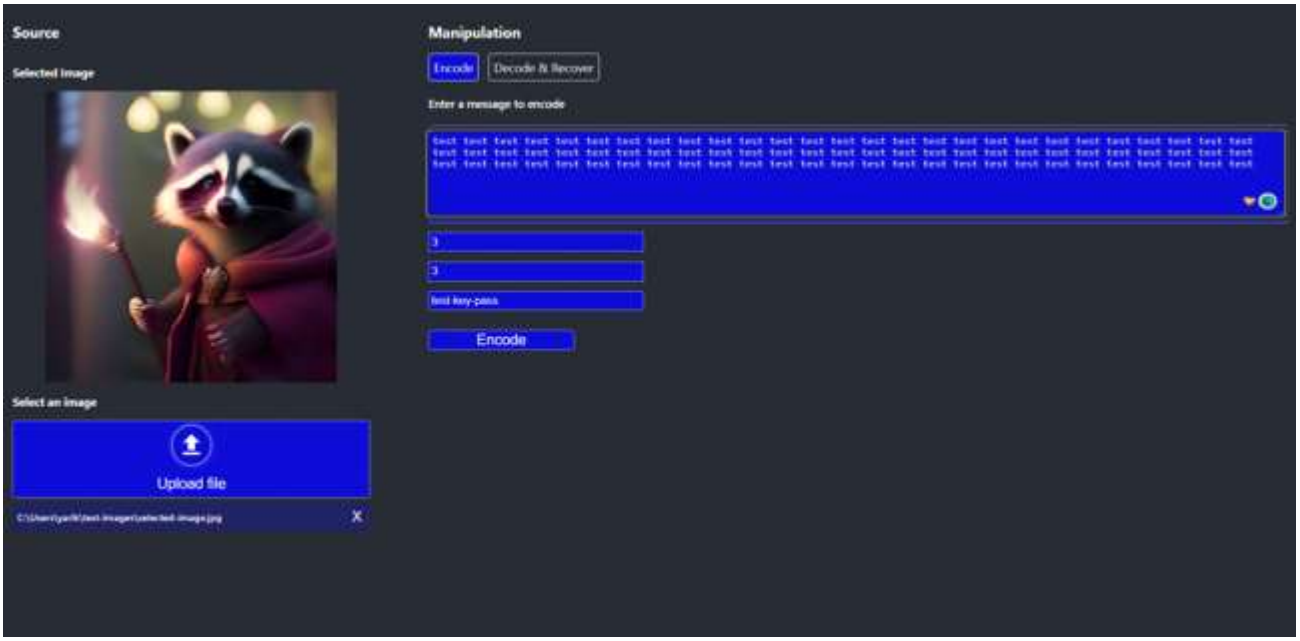



Рисунок 4.1 – Головна сторінка застосунку

Після того як обереться файл-контейнер, він буде відображений у спеціальному місці. Далі треба надати необхідні налаштування та натиснути кнопку “Encode”. У результаті буде отримано декілька файлів, які містять розподіли Шаміра. Приклад завантажуваних файлів показан на рисунку 4.2

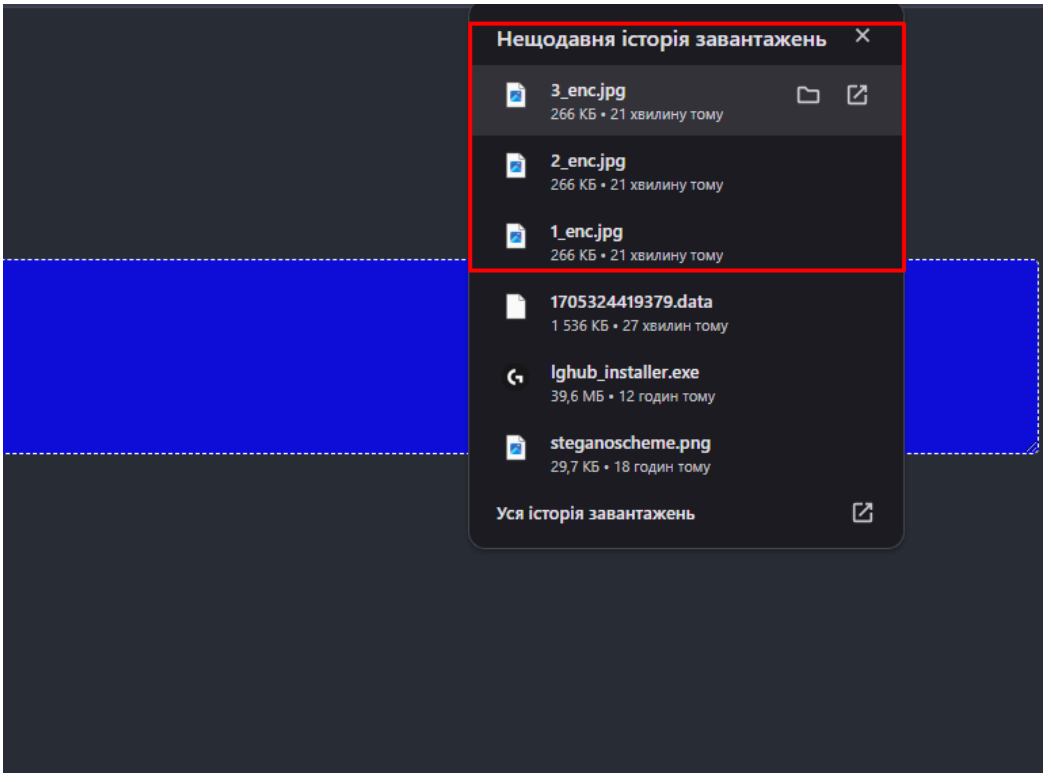



Рисунок 4.2 – Результат вкраплення повідомлення

Для вилучення повідомлення треба перейти на вкладку “Decode & Recover” та всі файли з розподілами Шаміра у лівій частині меню. Після чого треба ввести ключ необхідний для отримання повідомлення і натиснути кнопку “Decode & Recover”. Після чого у відповідному полі “Your message data” буде відображено усі розподіли Шаміра, які знаходяться у файлах.

У полі “Your recovered data” можна побачити відновлене повідомлення, вилучене зі стеганоконтейнерів та пересвідчитися що воно повністю співпадає.

Результат роботи вилучення можна побачити на рисунку 4.3.

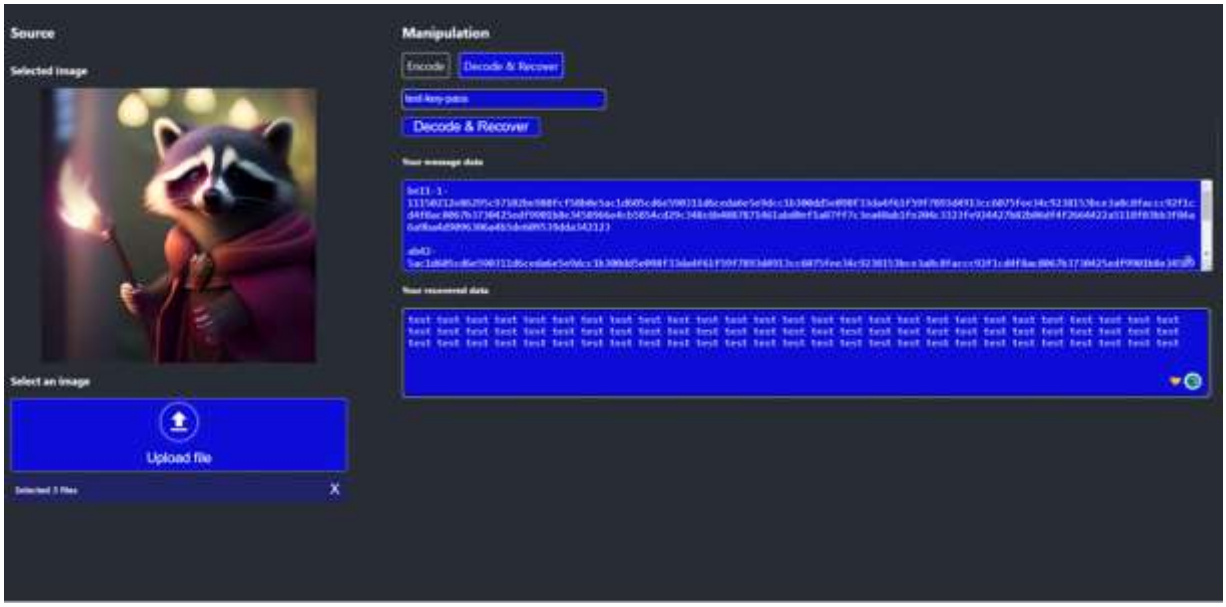



Рисунок 4.3 – Результат вилучення повідомлення

## 4.5 Аналіз результатів роботи web-застосунку

Для оцінки ефективності web-застосунку будуть застосовані два методи стеганоаналізу: хі-квадрат та RS. Ці методи аналізують піксельні області на предмет статистичних відхилень, що дозволяє їм ефективно виявляти вбудовані повідомлення, особливо коли використовується традиційний метод вставки найменш значущих бітів. У дослідженні буде використаний стеганоконтейнер, до якого застосовано метод розподілу секрету, розроблений Шаміром.

Атака методами хі-квадрат і RS дозволяє оцінити кількість інформації, прихованої в контейнері. Результати атак наведено на рисунках 4.4 та 4.5. Для проведення атак було використано онлайн-ресурс lsbtools. Щоб атакувати контейнери (зображення із вкрапленим повідомленням), треба перейти за посиланням https://desudesutalk.github.io/lsbtools та обрати необхідні стеганоконтейнери.

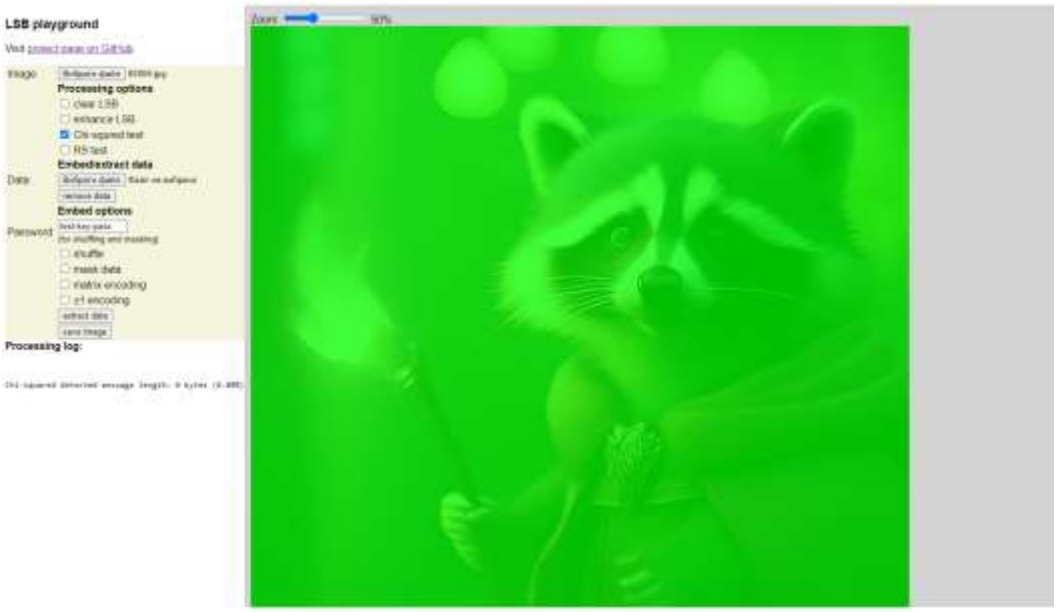



Рисунок 4.4 – Результат “Хі квадрат” атаки

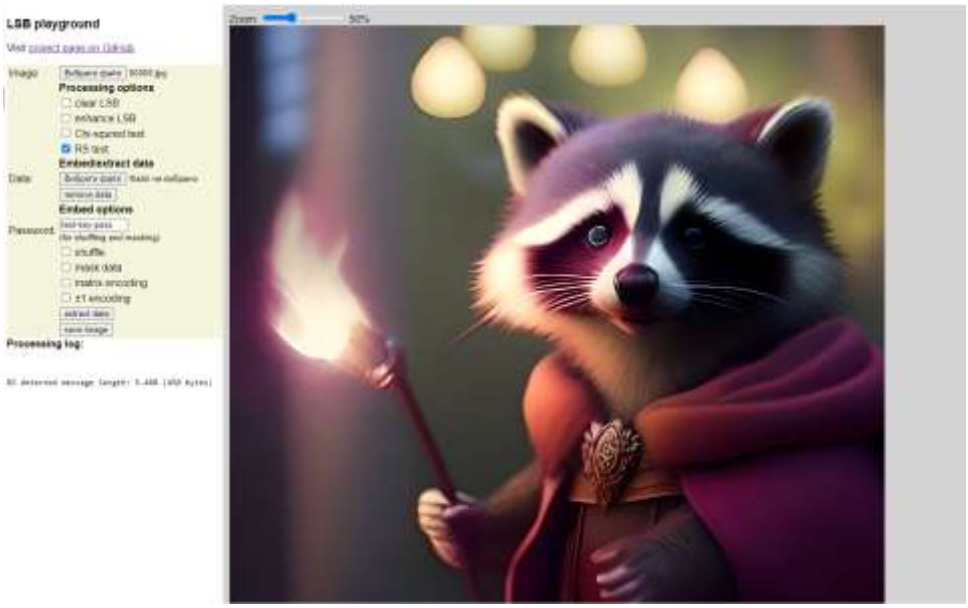



Рисунок 4.5 – Результат “RS” атаки

Можливо вказати, що результати атак є помилковими. Аналіз за методом хі-квадрат не виявив вбудованої інформації, тоді як RS-атака ідентифікувала лише 5,48% даних від загального об’єму контейнера, хоча на ділі в ньому містилося 18% від його розміру. Застосування випадкової послідовності при

вбудовуванні ускладнює процес виявлення прихованих даних для аналітиків. Це обумовлено неможливістю встановлення певної закономірності впровадження, яка б зазвичай використовувалася для детекції інформації під час атак. У таблицях 4.1 та 4.2 представлені результати RS-атаки та хі-квадрат атаки на стеганоконтейнери, до яких була додана інформація за допомогою стандартних та модифікованих методів НЗБ, залежно від ступеня заповнення контейнера. Точність атак на контейнери, а саме середнє співвідношення виявленого обсягу інформації до фактичного, зменшується з ростом заповнення контейнера. Для дослідження було обрано 50 контейнерів, в які проводилося вбудовування інформації.

Таблиця 4.1 – Результати точності хі-квадрат атаки

| Наповненість контейнера, % | НЗБ метод, % | Розроблений метод, % |
|---|---|---|
| 0 | 0 | 0 |
| 20 | 62.13 | 45.6 |
| 40 | 88.5 | 48.8 |
| 60 | 91.33 | 85.36 |
| 80 | 98.6 | 95.55 |
| 100 | 100 | 100 |

Таблиця 4.2 – Результати точності RS атаки

| Наповненість контейнера, % | НЗБ метод, % | Розроблений метод, % |
|---|---|---|
| 0 | 0 | 0 |
| 20 | 45.1 | 6.01 |
| 40 | 72.5 | 51.12 |
| 60 | 95.8 | 82.36 |
| 80 | 99.3 | 96.87 |
| 100 | 100 | 100 |

Використання схеми розподілу секрету Шаміра значно збільшує часову складність алгоритму відновлення початкового повідомлення. Це пов'язано з тим, що для відновлення використовується поліном Лагранжа, який є криптографічною функцією, що ускладнює процес відновлення.

Схема розподілу секрету Шаміра розбиває початкове повідомлення на кілька частин, які зберігаються в різних місцях. Це робить неможливим відновлення повідомлення без всіх частин, навіть якщо одна або кілька частин будуть втрачені або зловмисно змінені.

Таким чином, використання схеми розподілу секрету Шаміра підвищує стійкість контейнера до стеганоаналізу. Зловмисник, який отримає контейнер з вилученим повідомленням, не зможе відновити його до початкового вигляду без інших частин.

# ВИСНОВКИ

У даній роботі створено новий стеганографічний алгоритм, що включає модифіковане вкраплення НЗБ та схему Шаміра, збільшуючи місткість та безпеку контейнера. Також розроблено веб-додаток на .NET і React.

У початковій фазі розробки аналізувались різні методи стеганографії, їх особливості та ефективність. Вивчались методики створення та вкраплення даних у контейнери, а також шляхи підвищення їх безпеки та місткості. Результатом досліджень стало використання двох алгоритмів, включаючи модифікований, для оптимізації стеганоконтейнера. Модифікація алгоритму з випадковим порядком вкраплення ускладнює аналітикам виявлення прихованих даних через відсутність чіткої закономірності.

У результаті експериментальних досліджень було показано, що розроблений алгоритм має високу пропускну здатність та стійкість до стеганоаналізу. Зокрема, точність RS-атаки та атаки хі-квадрат на стеганоконтейнери, в які було впроваджено інформацію за допомогою розробленого алгоритму, значно нижча, ніж на контейнери, в які було впроваджено інформацію за допомогою традиційного алгоритму НЗБ.

На основі вищенаведеного можна зробити висновок, що в результаті проведеної роботи було досягнуто поставлену мету. Розроблений алгоритм поєднує в собі високу пропускну здатність та стійкість до стеганоаналізу, що робить його перспективним для використання в різних галузях.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Кошкіна Н.В. Методи стеганоаналізу з навчанням та класифікацією за характеристичними векторами / Н.В. Кошкіна // Праці міжнар. конф. “Питання оптимізації обчислень-XL”. Київ: Ін-т кібернетики ім. В.М. Глушкова НАН України. – 2015. – С. 153-154.

2. Mallat S. A Theory For Multiresolution Signal Decomposition: The Wavelet Representation /S.A. Mallat // IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989. – Vol. 11. – P. 674-693.

3. Кошкина Н.В. Стеганоаналіз цифрових зображень із застосуванням контрольного вкраплення / Н.В. Кошкина // Матеріали з Міжнар. наук.-техн. конф. «Захист інформації і безпека інформаційних систем», 5-6 черв. 2014. – Львів: Львівська політехніка, 2014. – С. 98-100.

4. Поліновський В.В. Інформаційна технологія для досліджень методівстеганографії і стеганоаналізу / В.В. Поліновський, В.Ю. Корольов, В.А. Герасименко, М.Л. Горинштейн // Комп’ютерно-інтегровані технології: освіта, наука, виробництво. – 2011. – №5. – С. 236-242.

5. Manjula Devi T.H. Detecting original image using histogram, DFT and SVM / T.H. Manjula Devi, H.S. Manjunatha Reddy, K.B. Raja, K.R. Venugopal, L.M. Patnaik // Intern. journal of recent trends in engineering. – 2009. – Vol. 1, №1. – P. 367-371.

6. Романчук Р.О. Вплив стеганографії та схеми розподілу секрету зображень на безпеку криптографічного ключа / Р.О. Романчук, А.О. Поліщук // Матеріали міжнародної наукової конференції «Актуальні наукові дослідження в сучасному світі», 26-27 грудня 2017 р. – С. 27-33.

7. Zadiraka V. Spectral methods of computer steganography problem decision / V. Zadiraka, N. Koshkina // Methods of effective protection of information flows /ed. by V. Zadiraka, Y. Nykolaichuk. – Ternopil: Ternograf, 2014. – P. 96-120.

8. Кошкіна Н.В. До питання часо-частотного аналізу сигналів в задачах комп'ютерної стеганографії / Н.В. Кошкіна // Праці міжнар. конф. "Питання оптимізації обчислень-XXXVI. Київ: Ін-т кібернетики ім. В.М. Глушкова НАН України. – 2011. – Том 1. – С. 301-355.

9. Мельник С.В. Світові тенденції розвитку цифрової стеганографії в контексті завдань за-безпечення інформаційної безпеки держави / С.В.Мельник, С.В.Кондакова // Актуальні проблеми управління інформаційною безпекою держави : зб. матер. наук.-практ. конф. – К. : Наук.-вид. відділ НА СБ України, 2010. – С. 134-138.

10. Shapiro J. Embedded Image Coding Using Zerotrees Of Wavelet Coefficients / / IEEE Transactions on Signal Processing, 1993. – Vol. 41, No. 12.

11. Said A., Pearlman W. A New Fast And Efficient Image Codec Based On Set Partitioning in Hierarchical Trees / / IEEE Transactions on Circuits and Systems for Video Technology, 1996. – Vol. 6. – P. 223-250.

12. Кошкина Н.В. Стеганоанализ бесключевых стеганосистем на основе атаки контрольным внедрением / Н.В. Кошкина // Междунар. научно-техн. журнал «Проблемы управления и информатики». – 2014. – № 6. – С. 137-144.

13. Кошкіна Н.В. Інформаційно-теоретична модель безпеки стеганографічних систем / Н.В. Кошкіна // Поступ в науку. – 2011. – №6, Т.1. – С.107-120.

14. Voloshynovskiy S.V. Visual communications with side information via distributed printing channels: extended multimedia and security perspectives / S.V. Voloshynovskiy, O. Koval, F. Deguillaume, T. Pun // Proc. of SPIE: Security, 93

Steganography, and Watermarking of Multimedia Contents VI, San Jose, USA, January 2004. – P. 408-445.

15. Suresh A. Image Texture Classification using Gray Level Co-Occurrence Matrix Based Statistical Features / A. Suresh, K.L. Shunmuganathan // European Journal of Scientific Research. – 2012. – Vol.75, № 4. – P. 501-597

16. Швідченко І.В. Аналіз програмного забезпечення зі стеганоаналізу / І.В. Швідченко //Искусственный интеллект. – 2012. – №3. – С. 457-494.

17. Яне Б. Цифровая обработка изображений / Б. Яне. – М.: Техносфера, 2007. – 583 с

18. Конахович Г.Ф. Компьютерная стеганография. Теория и практика / Г.Ф. Конахович, А.Ю. Пузыренко. – К.: МК-Пресс, 2006. – 288 с.

19. Хорошко В.А. Введение в компьютерную стеганографию / В.А. Хорошко, М.Е. Шелест. – Киев: Національний Авіаційний Університет, 2002. – 152 с.

20. Воробьев В.И. Теория и практика вейвлет-преобразования / В.И. Воробьев, В.Г. Грибунин. – СПб: ВУС, 2009. – 325 с.

21. Конахович Г.Ф. Компьютерная стегано-графия. Теория и практика / Г.Ф.Конахович, А.Ю.Пузыренко. – К. : МК-Пресс, 2006. – 288 с.

22. Столлингс В. Криптография и защита сетей: теория и практика. М: Вильямс. – 2001. – 855 с

23. Шнайер, Брюс. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си .– М.: Издательство ТРИУМФ, 2002 . – 526 с.

24. Защелкин К.В. Решение проблемы классификации блоков контейнера при jpeg-атаке на стеганографический метод Бенгама-Мемона-Эо-Юнг / К.В. Защелкин, А.А. Ищенко, Е.Н. Иванова // Радіоелектронні і комп’ютерні системи. – 2014. – № 6 (70). – С. 164-168.