**Міністерство освіти і науки України**
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: Порівняльний аналіз стеганографічних алгоритмів приховування інформації в зображеннях

Виконав(ла): студент(ка) VI курсу, групи СПм-61
спеціальності 121 «Інженерія програмного забезпечення»
(шифр і назва спеціальності)

_____ Резнік Д.В.
(підпис) (прізвище та ініціали)

Керівник _____ д.ф.-м.н., професор Петрик М.Р.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ к.ф.-м.н, доцент Бойко І.В.
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ д.ф.-м.н., професор Петрик М.Р.
(підпис) (прізвище та ініціали)

Рецензент _____ к.т.н, доцент Баран І.О.
(підпис) (прізвище та ініціали)

Тернопіль

2020

# АНОТАЦІЯ

Кваліфікаційна робота на тему «Порівняльний аналіз стеганографічних алгоритмів приховування інформації в зображеннях».

Резнік Дмитро Володимирович Тернопільський національний технічний університет імені Івана Пулюя, Факультет комп'ютерно-інформаційних систем і програмної інженерії, Кафедра програмної інженерії, група СПм–61, Тернопіль, 2020.

Кваліфікаційна робота містить: 96 с., 21 рис., 16 табл., 27 посилань.

Зважаючи на різноманіття стеганографічних алгоритмів є достатньо складно виконати вибір того чи іншого методу для приховування секретної інформації в зображеннях. В роботі поставлено завдання виконати порівняльний аналіз стеганографічних алгоритмів на основі заданих критеріїв та умов використання.

Метою даної роботи є розробка комбінованого стеганографічного алгоритму приховування інформації в зображеннях на основі результатів попереднього порівняльного аналізу.

Під час виконання роботи проведено дослідження переваг та недоліків сучасних стеганографічних алгоритмів приховування інформації у просторовій і частотній областях, зібрано статистичні дані кількісних характеристик обраних алгоритмів та виконано їх порівняльний аналіз методом аналізу ієрархій.

На основі даних виконаного аналізу запропоновано модифікацію одного з стеганографічних алгоритмів з використанням завадостійкого кодування.

Реалізація алгоритмів проводилась з використанням мови програмування C#, фреймворка .NET Core 2.2, та бібліотеки обробки зображень SixLabors.

Результати проведеного дослідження, аналізу та реалізації вдосконаленого алгоритму можуть використовуватись для передачі конфіденційних даних по відкритих каналах зв’язку або подальшої роботи над вдосконаленням алгоритмів приховування інформації в зображеннях.

СТЕГАНОГРАФІЯ, СТЕГАНОКОНТЕЙНЕР, ПРОСТОРОВА ОБЛАСТЬ, ЧАСТОТНА ОБЛАСТЬ, СПОТВОРЕННЯ, ЗАВАДОСТІЙКЕ КОДУВАННЯ.

# ABSTRACT

Master's thesis on “Comparative analysis of steganographic algorithms of hiding information in images”.

Reznik Dmytro Volodymyrovich, Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, SPm-61 Group, Ternopil, 2020.

Certification work contains: 96 p., 21 fig., 16 tables, 27 references.

Considering a variety of steganographic algorithms it is hard to pick the best one for hiding secret information. The task of this thesis is to evaluate those algorithms by some predetermined criteria and its usage circumstances.

The goal of this thesis is to create a combined steganographic algorithm of information hiding in an image based on the result of a previous analytical analysis.

During work on this thesis, a research of advantages and disadvantages of modern steganographic algorithms of hiding information in a special and frequency domain was made, statistical data of qualitive characteristics of chosen algorithms was gathered and analytical analysis using analytic hierarchy process was made.

Based on the analysis data a modification of one of the steganographic algorithms with a use of noise-tolerant coding was proposed.

The project was created using a C# programming language, a framework .NET Core 2.2, and a library for image processing SixLabors.

The results of a research, analysis and implementation of the advanced algorithm could be used for confidential data transfer through an insecure channel and further work on usage of algorithms of hiding information in images.

STEGANOGRAPHY, STEGANOCONTAINER, SPATIAL DOMAIN, FREQUENCY DOMAIN, DISTORTION, NOISE-TOLERANT CODING.

**ЗМІСТ**

# ПЕРЕЛІК СКОРОЧЕНЬ

AD – Absolute Difference

JPEG – Joint Photographic Experts Group, формат зображення

LSB – Least significant bit

MD – Maximum Difference

NC – Normalized Cross-Correlation

RGB – red green blue, колірна модель

SNR – Signal Noise Ratio

ВДТ – візуальний дисплейний термінал

ДДКМ – метод Дамстедгера-Делейгла-Квіксвотера-Мака

ДКП – дискретне косинусне перетворення

ЕОМ (ПЕОМ) – електрообчислювальна машина (персональна)

IF – Image Fidelity

МАІ – метод аналізу ієрархій

НЗ фільтр – низькочастотний фільтр

НЗБ – найменший значущий біт

ПВЧ – псевдовипадкові числа

СМ – стеганографічний метод

# ВСТУП

Становлення стеганографії ключовою частиною окремих галузей виробництва та існування суспільства сформувалось вже дуже давно, оскільки питання безпеки інформації завжди було важливим в усі часи. Проте більш детальніше питання захисту інформації було переглянуте з поширенням такого явища, як обмін конфіденційними даними в мережі Інтернет.

Одними із напрямків науки, що мають на меті запобігання доступу до приватної інформації, є криптографія і стеганографія. У випадку, коли криптографія забезпечує захист даних, факт наявності методів захисту в яких є відомим, то стеганографія вивчає методи прихованої передачі інформації по відкритих каналах зв'язку без явного вказання того чи іншого захисту корисної інформації. Завдання стеганографії можна сформулювати як організацію передачі секретних повідомлень таким чином, щоб зміст повідомлення та факт його передачі були відомі тільки для зацікавлених сторін.

Існують декілька видів стеганографії та приховуванням інформації. Для прикладу, захистом цифрових даних займаються комп'ютерна і цифрова стеганографія, які вважаються належними до класичного напрямку. Основою цифрової стеганографії є те, що приховування або вбудовування додаткової інформації в цифрові об'єкти відбувається так, що завдані спотворення знаходяться нижче порогу чутливості пересічної людини. В свою чергу комп'ютерна стеганографія спеціалізується на приховуванні даних в різні бінарних даних так, щоб стандартними способами читання було неможливо виявити їх наявність.

Варто зазначити вчених, які досліджували ефективні алгоритми приховування інформації в своїх працях, а саме: G. J. Simmons, J. Fridrich, R. J. Anderson, W. Bender, N. Morimoto, I. Pitas. Необхідно зауважити, що тривалий час практично не було вітчизняних вчених, які би досліджували ті чи інші аспекти стеганографії, проте із запровадженням державної спеціальної політики в галузі безпеки (Закон України "Про основи національної безпеки України") кількість публікацій та досліджень зростає. Це передбачено необхідністю захисту державних

локальних мереж, передбачення загроз витоку інформації, поширення недостовірної інформації. Значний вклад у розвиток стегоаналізу внесли такі вітчизняні науковці: В. К. Задірака, В. А. Хорошко, М. Є. Шелеста, Г. Ф. Конахович. Серед праць в галузі стеганографії зазначимо кілька магістерських робіт студентів ТНТУ, а саме: Сачик Т.В. [16], Гулка Ю.І. [17], Кінах Я. І. [18], Чертова М. [19].

Більшість робіт вище перелічених науковців спрямована на розробку та вдосконалення методів приховування інформації, та все ще актуальною є проблема вибору оптимального алгоритму для специфічних цілей. Також актуальність зазначеної проблеми обумовлена відсутністю універсальних рішень, що змогла б покрити широкий спектр застосування стеганографії. Так як зображення є одним з найпоширеніших видів інформації після тексту, то доцільно є те, що приховування конфіденційних даних в різних форматах зображення буде слушним в багатьох випадках. При використанні різноманітних стеганографічних методів приховування інформації необхідно враховувати можливе пошкодження якості зображення, що буде ідентифікатором для виявлення зоровою системою людини пошкоджень оригіналу. Отож, виконання порівняльного аналізу стеганографічних методів приховування інформації, дослідження їх переваг та недоліків їх використання саме на зображеннях, реалізація певного модифікованого методу, що дозволить зменшити пошкодження якості зображення і забезпечити точність передачі вбудованого повідомлення, є актуальною задачею.

Метою даної роботи є розробка комбінованого алгоритму приховування інформації в зображеннях на основі існуючих методів з врахуванням результатів порівняльного аналізу на основі попередньо зазначених критеріїв.

Щоб досягнути поставленої мети, потрібно вирішити наступні задачі:

1. Дослідити сучасні стеганографічні алгоритми щодо їх переваг і недоліків та виконати порівняльний аналіз згідно обраних критеріїв.

2. Вдосконалити алгоритм на основі даних проведеного порівняльного аналізу.

3. Практично реалізувати вдосконалений алгоритм методами обраної мови програмування.

Під час написання роботи було проведено дослідження сучасних стеганографічних алгоритмів на предмет переваг та недоліків та виконано їх порівняльний аналіз.

Об’єкт дослідження – процес приховування інформації в зображеннях.

У роботі використовувались наступні методи дослідження: експеримент та вимірювання (програмна реалізація стеганографічних методів та вимірювання кількісних значень відповідних критеріїв), аналіз та синтез (аналіз отриманих кількісних показників критеріїв та виконання порівняльного аналізу на основі синтезу отриманої інформації), моделювання (блок-схеми, для програмної реалізації комбінованого методу приховування інформації).

Науковою новизною є розроблений модифікований стеганографічний алгоритм на основі проаналізованих даних порівняльного аналізу з використанням широко вживаного алгоритму Коха-Жао приховування інформації в частотній області зображення, що дозволило покращити стійкість стегоконтейнера до модифікацій та спотворень спричинених шумом.

Практичною цінністю одержаних результатів є можливість передачі корисної інформації, вбудувавши у частотний домен зображення завдяки алгоритму Коха-Жао та підвищення надійності передачі (покращення стійкості до втрати даних) завдяки застосуванню коригувального коду Хеммінга.

Основні положення і результати роботи були представлені на VIII науково-технічній конференції «Інформаційні моделі, системи та технології».

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

## 1.1 Історичний екскурс в розвиток стеганографії

Стаганографія як наука зародилась у Єгипті, який більшість називає її колискою, хоча найдавніші “стеганографічні повідомлення” були простими наскальними малюнками тодішніх народів.

Також вважають, що першим у літературі про стеганографічні методи згадував Геродот, коли описував процес передавання повідомлень Демарата про можливий напад на Грецію з використанням воскових дощечок. Для цього Демарат зняв захисне воскове покриття з дошок, написав текст повідомлення про загрозу на дереві, а потім знову покрив дошки воском. Ще один епізод, що датується тим самим часом, це передача повідомлення за допомогою голови раба. Щоб передати таємне повідомлення, голову раба голили, татуювали на шкірі, чекали поки волосся знову відросте і відправляли назад з повідомленням.

У Середньовіччі народження інквізиції потягнуло за собою стрімкий розвиток криптографії та стеганографії. Варто зазначити відмінність між криптографією та стеганографією. При використанні криптографічних методів, отримувач (зловмисник чи адресат) знатиме точно чи отримане повідомлення є зашифрованим, оскільки буде потрібно ввести додатково секретний (криптографічний) ключ чи пароль для отримання корисного повідомлення. На противагу цьому завдяки стеганографічним методам секретна інформація вбудовуються в звичайне повідомлення і лише отримувач знає про наявність секретного та методу його отримання.

Саме в Середньовіччі було започатковано комбіноване використання криптосистем (шифрів) та стеганографічних методів. Чернець Трітеміус (1462-1516) у XV столітті описав безліч різних способів таємної передачі повідомлень, а у 1499 році інформацію з цих документів систематизували та об’єднали у книгу “Steganographia”. Саме Іоан Трітеміус вважається основоположником терміну “стеганографія”.

У 17-18 століттях впроваджувались спеціальні установи, так звані “чорні кабінети”, ціллю яких було перехоплення, перегляд та розшифрування інформації у листах. До працівників даних установ можна віднести крім криптографів та дешифрувальників ще й інших спеціалістів, зокрема хіміків. Їх вклад в справу був вагомим, адже в листуванні активно використовувались невидимі фарби. Прикладом можна зазначити історичний епізод за участі агента кардинала Мазаріні та монаха Берто, якого заарештували повсталі дворяни в місті Бордо у Франції. Ув’язненому дозволили написати листа священику, в кінці якого з релігійним змістом чернець написав приписку з порадою використати мазь для проявлення невидимих чорнил. Контекст приписки ніхто не зміг зрозуміти і лист без завад прийшов до адресата, після цього монаха врятували.

Також випадки використання стеганографічних методів спостерігались під час завоювання півдня сіверянами. Агенти Семюель Вудхулл та Роберт Тоунсенд успішно відправили Джорджу Вашингтону у 1779 році лист з прихованою інформацією за допомогою невидимих чорнил.

На початку 20 століття соціалісти-революціонери користувались різними специфічними чорнилами, що відобразилось в літературі. Яскравим прикладом є повість “Біля витоків майбутнього”, де описано використання молока як специфічного чорнила для приховання та передачі секретних повідомлень. Проте державна гвардія також вже на той час була знайома з таким методом (архів містить документ, який описує, як використовувати відповідні чорнила), тому повідомлення було перехоплене у есерів.

Вагомого поштовху у розвитку стеганографії спричинило використання фотографічних мікроточок. Перші згадки про такий стеганографічний метод датуються часами франко-прусської війни (в 1870 р.), що завдало чималих проблем американським спецслужбам в часи Другої світової війни. Направду метод є достатньо простим, адже мікроточка повинна бути розміром як друкарська крапка, а по факту була мікрофотографією корисної інформації (на ті часи передавали навіть фото цілих сторінок тексту, креслення). Зазначимо, що такий метод дозволяв передавати надзвичайно великі обсяги інформації.

На даний момент стеганографія широко застосовується для захисту виключного права, недекларованого зберігання інформації, захисту авторського права, достовірності інформації, реалізації стеганографічних botnet-мереж. Розглянемо найбільш цікаві та практично реалізовані приклади.

Компанія Amazon для захисту файлів формату mp3 додавала спеціальний ідентифікатор, який відображав інформацію щодо локації завантаження та додаткову інформацію про власника. У випадку, якщо дві різні людини завантажували одну і ту ж композицію, файли за розміром були ідентичними, проте при побутовому порівнянні – зовсім різними.

В стеганографічних мережах використовується поняття прихованої передачі управляючого сигналу, що дозволяє отримувати відповідну інформацію та зчитування її спеціально запрограмованим ботом.

### 1.2 Ключові поняття в стеганографії

Стеганографією вважають науку про методи захисту інформації шляхом приховування факту її існування в певному середовищі. Основним завданням є приховування факту існування вбудованого (секретного) повідомлення з метою його ж подальшого захисту від зловмисників. Для забезпечення реалізації використовується наявні різноманітні технічні, хімічні, фізичні і психологічні методи такого приховування [1].

Класичним завдання стеганографії називають організацію передавання секретного (вбудованого) повідомлення таким чином, щоб зміст повідомлення та факт його передачі були приховані від усіх, крім наперед визначених одержувачів. З метою вирішення цього завдання використовують стеганоконтейнер, в яке вбудовують секретне повідомлення, і яке по суті є найбільш необхідним для передачі. Таким чином, при розробці стеганографічних методів повинна виконуватись умова забезпечення прозорості переданих конфіденційних даних завдяки зміні певної кількості бітів. Зауважмо, що при цьому інформація в

контейнері не повинна призводити до особливих спотворень якості (не повинно бути візуально видимих артефактів).

Цифрові фотографії, текст, музика, відео вважають найпоширенішими контейнерами. Наприклад, при використанні графічних файлів як контейнера для сторонніх спостерігачів чи одержувачів, процес обміну повідомленнями буде розглядатися як звичайний обмін файлами. Необхідно зауважити одну важливу умову: не повинно бути відкритого доступу до вихідного файлу, вибраного як контейнер, так до файлу із прихованим повідомленням. В разі відкритого доступу звичайне порівняння файлів дозволить одразу виявити наявність секретного повідомлення.

Існують наступні галузі стеганографії: класична стеганографія, комп'ютерна, цифрова [10]. Зауважимо також, що існують менш широко вживані, а саме: лінгвістична та квантова стеганографія. Кожна з них є доповненням до класичної стеганографії та сягає глибокої давнини.

Хімічні та фізичні методи також входять до класичної стеганографії. Зазвичай хімічні методи стеганографії є представленими завдяки використанню симпатичних (невидимих) чорнил: симпатичні хімікалій і органічних рідин. Симпатичні хімікалії – один із найпоширеніших методів класичної стеганографії. Розглянемо коротко процес запису з використанням хімічних методів. Для прикладу, на перший шар наноситься важливий запис невидимим чорнилом, а на другий шар – незначущий запис вже видимими чорнилами. Лише за певних умов текст написаний таким чином з'являється, це може бути нагрівання, освітлення, проявлення хімічними речовинами. Органічні рідини мають властивості, подібні до симпатичних хімікалій: при нагріванні вони темніють (містять велику кількість вуглецю).

Методи класичної (традиційної) стеганографії зображені на рисунку 1.1. Як представлено на рисунку 1.1 до фізичних методів належать схованки, методи камуфляжу, мікроточки (мікрокрапки), голограми. Найбільший інтерес становить різноманітні носії інформації. Оскільки, при застосуванні певного стеганографічного методу до них, приховане повідомлення буде неможливо

прочитати стандартними методами читання/запису. Особливо варто відзначити стандартні носії інформації, засоби обчислювальної, аудіо- та відеотехніки.

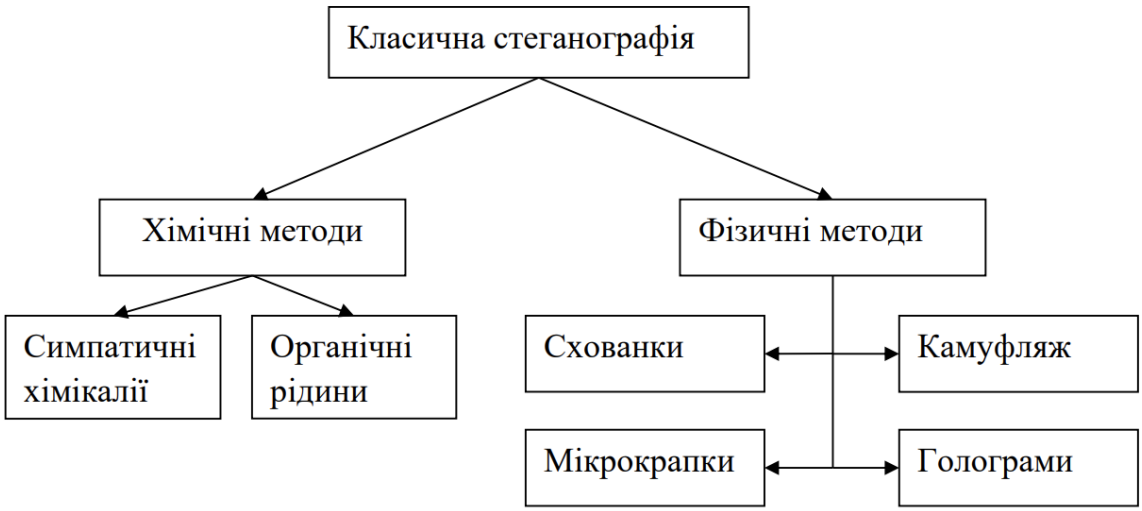



Рисунок 1.1 – Методи класичної (традиційної) стеганографії

Поряд із тим з’являються різноманітні нові технології, які на основі традиційної стеганографії використовують найновіші досягнення мікроелектроніки (голограми). Зауважмо, що даний метод володіє найбільшою стійкістю до спотворення. Голографічний метод полягає в тому, що в контейнер вбудовуються не безпосередньо конфіденційні дані, а їх голограма. При цьому такий підхід дозволяє вбудовувати секретні дані у звичайні фотографії на папері або пластику. Основним недоліком цього методу є обмежений обсяг вбудованих даних. Найдоцільніше використовувати голографічний підхід для маскування невеликих зображень, відновлення яких дозволяє незначну втрату якості: зразки підписів, відбитки пальців.

Ідея методу камуфляжу полягає у маскуванні секретного повідомлення для “злиття” з кольором об'єкта, який виконує роль контейнера.

Комп’ютерна стеганографія бере свої корені від класичної стеганографії та оперує характеристиками різноманітних комп’ютерних платформ. Прикладом може стати вільна стеганографічна файлова система StegFS для Unix подібних

операційних систем. Іншим способом приховування інформації є запис повідомлення у нульову доріжку гнучких дисків. Загалом, даний розділ стеганографії займається записом даних на носії так, щоб за допомогою стандартних способів читання не було можливості дізнатись про їх наявність.

Метод приховування інформації у невикористаних місцях гнучких дисків полягає в тому, що вбудоване повідомлення записується на нульову доріжку. Недоліками такого методу є низька продуктивність, передача невеликих повідомлень, простота виявлення.

Розглянемо особливостей файлової системи: зберігання на жорсткому диску будь-якого файла (за винятком деяких FS, таких як ReiserFS) завжди займає цілу кількість кластерів. Наприклад, стандартний розмір кластера становить 4 КБ у файловій системі FAT32. Відповідно, для зберігання 2 КБ інформації на диску виділяється 4 КБ жорсткого диску, з них 2 КБ для корисної інформації, а інші два є пустими. Такі пусті обсяги пом’яті жорсткого диску можна використовувати для зберігання інформації в стеганографічних цілях. Недоліком цього методу є достатня простота виявлення.

Цифрова стеганографія (рис. 1.2) є одним із напрямів класичної стеганографії. Основна ідея полягає у тому, щоб приховати або вбудувати додаткову інформацію в цифрові об’єкти, при цьому викликаючи найменше спотворення. Зазвичай, такі спотворення, зважаючи на наявні формати цифрових об’єктів (зображення, відео, аудіо, текстури 3D), знаходяться нижче порогу чутливості людини, що призводить до непомітних для людського ока змін.

Лінгвістична стеганографія відображає методи приховування секретної інформації в непримітний текст, на основі застосування властивостей мови та лінгвістичних ресурсів. Умовне письмо та семиграми вважають основними категоріями лінгвістичних методи стеганографії (рис. 1.3).

Завдяки методам лінгвістичної стеганографії можливо передавати повідомлення великої довжини, що є істотною перевагою у порівнянні з іншими методами.


Рисунок 1.2 – Основні методи цифрової стеганографії

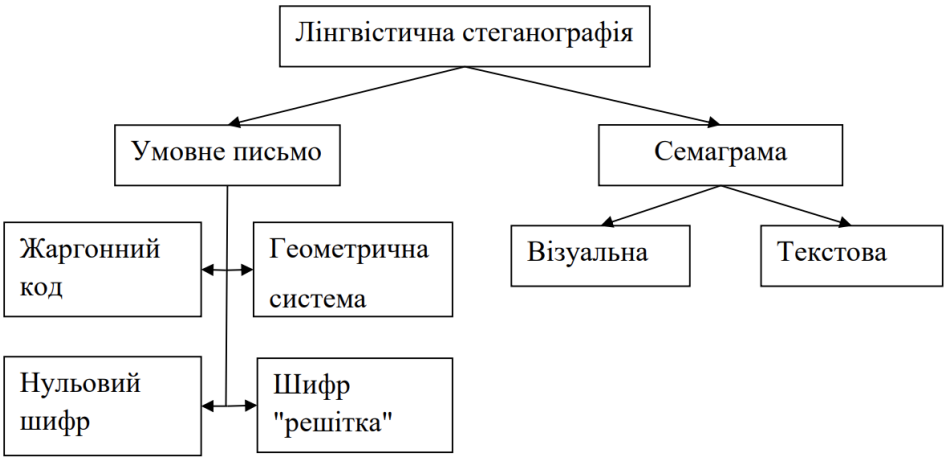



Рисунок 1.3 – Основні методи лінгвістичної стеганографії

Необхідно зазначити основні недоліки лінгвістичної стеганографії, а саме: можливість випадкового виявлення алгоритму кодування, тобто візуального виявлення на основі ідентифікації значної різниці між модифікованими та повідомленням-оригіналом); практична складність у реалізації.

Квантова стеганографія не є настільки популярною, проте існує ряд робіт у яких пропонується певні моделі систем захисту інформації, що використовують квантові властивості. За своєю суттю квантова стеганографія синтезує класичні та квантові обчислення на основі об’єднаних концепцій квантової фізики та класичної теорії інформації. На рисунку 1.4 представлено класифікацію методів квантової стеганографії.

Розглянемо типову стеганографічну систему (стеганосистема), що є сукупністю засобів і методів, які необхідні для формування прихованого каналу передачі даних [2].

За типом стегоключа стеганосистеми розподіляються на системи з секретним та з відкритим ключами.

У стеганосистемі з секретним ключем використовують тільки один ключ, завдяки якому здійснюють вбудовування і витягнення повідомлення.

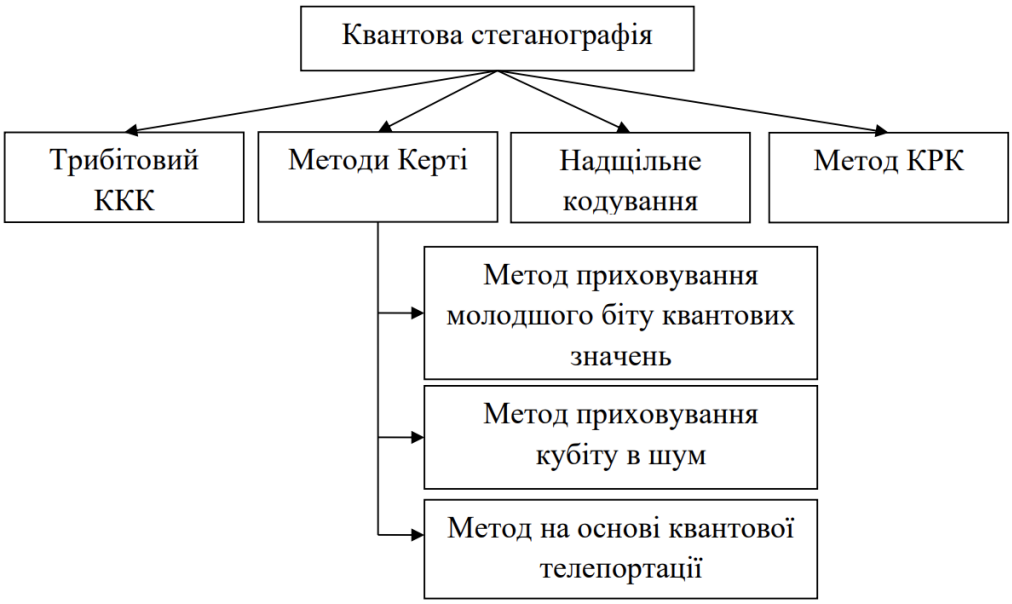



Рисунок 1.4 – Методи квантової стеганографії

У стеганографічній системі з відкритим ключем використовують різні ключі для вбудовування та отримання повідомлення. Особливістю такої пари ключів в тому, що знаючи один ключ неможливо вивести обчисленнями інший. Тому один

з них (відкритий ключ) можна передавати відкрити по незахищених каналах зв'язку.

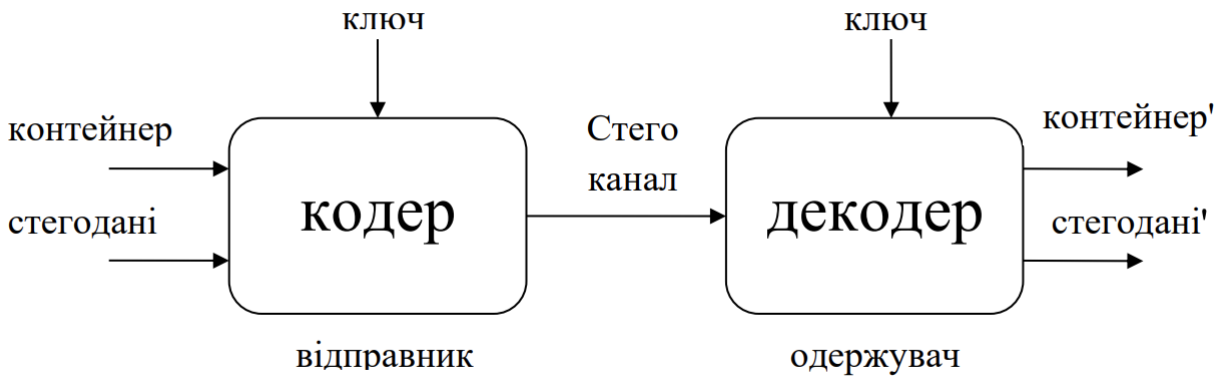



Рисунок 1.5 - Модель стегосистеми

Контейнером називають будь-яку інформацію, яка призначена для приховування секретних повідомлень [3].

Порожнім контейнером називають контейнер без будь-якого секртеного повідомлення. Заповненим контейнером або стегоконтейнером називають контейнер, що містить приховану інформацію [4].

Вбудованим (прихованим або секретним) повідомлення називають таке повідомлення, що є вбудованим в контейнер. Стегоключ або просто ключ – секретний ключ, необхідний для приховування інформації [2]. В залежності від рівня захисту ключів може бути кілька.

Стеганографічні методи можна класифікувати за багатьма ознаками.

За призначенням розрізняють: для захисту конфіденційних даних, для захист авторських прав та для автентифікація даних.

За принципом приховування повідомлень: у просторову область та у частотну область.

За форматом контейнера класифікують на текстові, аудіо, графічні та відео.

За типом контейнера існують потокові та фіксовані.

Кожна з перерахованих вище категорій характеризується певним співвідношенням між стійкістю до зовнішніх впливів і розміром самого вбудованого повідомлення.

Наявні стеганографічні методи характеризуються залежністю між надійністю системи та обсягом вбудованих даних, що представлено на рисунку 1.6.

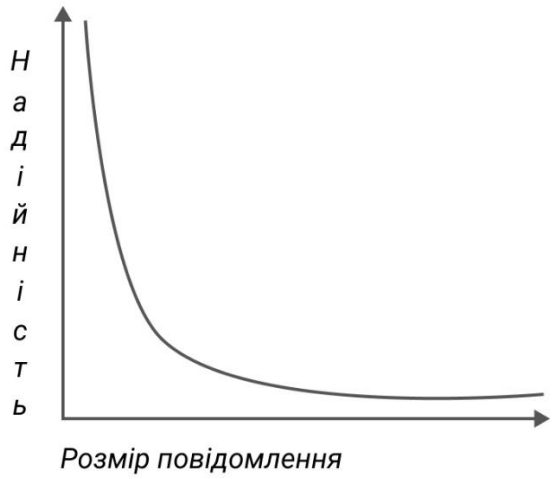



Рисунок 1.6 – Залежність надійності системи від обсягу вбудованих даних

Представлена залежність вказує на те, що при збільшенні кількості вбудованих даних страждає надійність системи (при однаковому розмірі контейнера).

### 1.3 Методи приховування інформації в графічних зображеннях

Сукупність стеганографічних методів (СМ) можна розділити на два класи на основі принципів, що використовуються при їх побудові, а саме не форматні та неформатні.

Неформатні СМ неминуче призводять до спотворення зображень, що спричиненні фактом приховування інформації і використовують безпосередньо первинні дані зображення. Зауважмо, що даний клас СМ є достатньо стійким до активних та пасивних атак.

Форматні СМ ґрунтуються на властивостях конкретного формату зображення, що використовується при передачі. Для їх реалізації виконується детальний аналіз формату зображення, його структури, з метою знаходження певних службових бітів, змінивши які не втратиться та не спотвориться зображення оригінал. У випадку загальновідомості стеганографічної системи дуже легко виявити факт приховування інформації, проте в протилежному випадку неможливо побудувати автоматичний алгоритм виявлення вбудованих даних. Це є недоліком при використання форматних СМ.

В даній роботі будуть розглядатись форматні СМ в частотній та просторовій області для приховування даних.

### 1.3.1 Методи вбудовування інформації в просторовій області графічного зображення

Методи приховування в просторовій області засновані на принципі заміни бітів зображення, що несуть надлишкову і малозначиму інформацію на біти секретного повідомлення.

Найпоширенішим та базовим (на основі якого будується більшість із цієї групи) методом є заміна найменш значущих бітів (НЗБ) послідовно розташованих пікселів зображення бітами секретної інформації. Зазвичай обсяг приховуваної інформації є меншим за обсяг всього зображення-оригіналу, тому після успішного виконання стеганографічного методу отримують дві області з різними статистичними властивостями, які можуть бути ідентифікованими статистичними методами стегоаналізу. З метою уникнення виявлення факту приховування, вбудоване повідомлення доповнюють доповнюють інформаційним сміттям – випадковими бітами. Обсяг такого інформаційного сміття обирають таким, щоб його бітова довжина дорівнювала кількості пікселів в зображенні. Зазначимо переваги використання цього методу: безсумнівна простота реалізації та значна корисна ємність контейнера. Недоліком є те, що при будь-якому спотворенні контейнера вбудована інформація також буде спотвореною.

Наведемо перелік найчастіше вживаних методів приховування в просторовій області графічного зображення [5]:

− заміни найменш значущого біта;

− псевдовипадкової перестановки;

− псевдовипадкового інтервалу;

− блочного приховування;

− заміни палітри;

− квантування зображення;

− Дамстедгера-Делейгла-Квіксвотера-Мака;

− Куттера-Джордана-Боссена.

Детально алгоритми приховування інформації в просторовій області зображення та відповідні пояснення наведено у розділі 2.

### 1.3.2 Методи вбудовування інформації в частотній області графічного зображення

Методи приховування в просторовій області нестабільні при стисненні з втратами, що спричиняє спотворення вбудованої інформації. Це є суттєвим недоліком при використанні таких методів. Проте він не поширюється на СМ приховування в частотній області.

Для представлення зображення в частотній області використовують декілька базисів, на основі яких виконується декомпозиція зображення. Серед найбільш вживаних варто зазначити дискретно-косинусне перетворення (ДКП), дискретне перетворення Фур'є (ДПФ), вейвлет-перетворення, перетворення Карунена-Лоева (метод головних векторів, розклад на власні вектори та значення). Такі перетворення застосовуються до окремих частин зображення, або до всього зображення.

Нехай маємо корисне повідомлення $W$, яке необхідно вбудувати у зображення. Відповідно частотна область зображення $S_0$ зміниться в залежності від повідомлення $W$. Наступним кроком є виконання зворотного перетворення, в

результаті якого отримаємо заповнений контейнер $S_W$. Для знаходження та отримання повідомлення $W$ необхідно виконати попередні кроки у зворотному порядку.

До приховування в частотній області можна віднести такі методи:

− метод Коха і Жао;

− метод Ху і Ву;

− метод Бенгама-Мемона-Ео-Юнг;

− метод Фрідріх.

Детально алгоритми приховування інформації в частотній області зображення та відповідні пояснення наведено у розділі 2.

### 1.4 Критерії якості стеганографічної системи

Трьома основними завданнями захисту інформації є забезпечення конфіденційності (confidentiality), цілісності (integrity) та доступності (availability) інформації. Ці три властивості інформації формують тріаду CIA. Оскільки стеганографія є одним із ключових елементів захисту інформації, на рівні із криптографією, налаштуванням апаратно-технічних засобів, виконанням інженерних та організаційних завдань, тому відповідно повинна також відповідати глобальним цілями захисту інформації і забезпечувати тріаду цілей CIA. Звичайно все залежить від поставленої мети, у випадку приховування передачі інформації, стеганографія повинна забезпечувати конфіденційність та цілісність вбудованої корисної інформації.

Обсяг конфіденційної інформації, що підлягає приховуванню, і стійкість цих даних до завад та шуму вимагають використання декількох методів приховування. Це спричинено тим, що кожен з методів приховування даних націлений на певну мету і не здатний досягти всіх цілей стеганографії одночасно. Відповідно з'являється необхідність у ефективному методі порівняння наявних алгоритмів та обрання найбільш оптимального для конкретного випадку. Перш за все, необхідно визначити основі цілі застосування стеганографії, а вже потім на їх основі

формувати критерії для обрання. Таким чином, крім виконання цілей CIA до інформаційної безпеки, до стеганографії додаються додаткові цілі, що формують особливу для неї тріаду, представлену на рис. 1.7, яка включає стійкість (robustness), місткість (capacity), непомітність (imperceptibility). Вперше ці вимоги були наведені та опубілковані в роботі Wang [11].

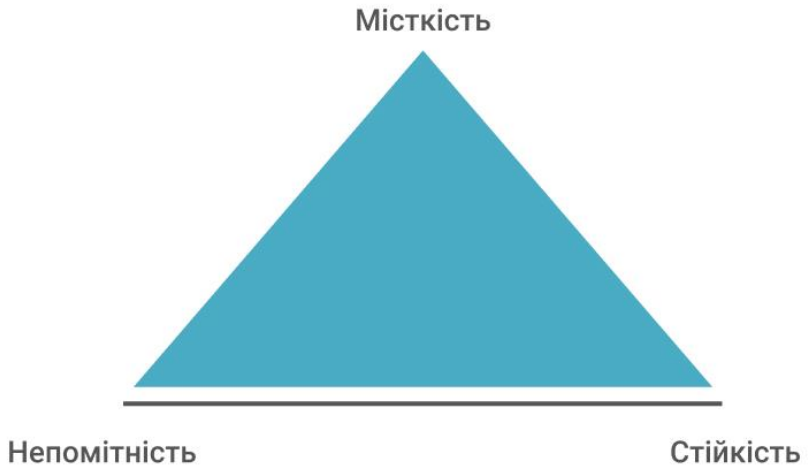



Рисунок 1.7 – Тріада вимог стеганографії [11]

Розглянемо детально кожен елемент із тріади вимог щодо стеганографії.

Під стійкістю розуміється можливість отримання секретної інформації із стегоконтейнера після навмисного спотворення контейнера переданого через канал зв'язку із завадами. Для визначення стійкості стеганографічних методів використовують атаки стегоаналізу, які будуть наведеними в наступному підпункті цього ж розділу.

Місткість це найбільша кількість інформації, яка може бути вбудована в стеганографічний контейнер. Перш за все, дана характеристика залежить від стеганографічного алгоритму та від властивостей контейнера. Представимо наявні метрики для вимірювання місткості: кількість бітів секретного повідомлення, що може бути вбудоване в піксель зображення; відношення розміру секретного повідомлення щодо максимального розміру повідомлення, яке може бути вбудовано в цей контейнер [12].

Властивість непомітності є забезпеченою у тому випадку, коли немає істотної різниці між стеганоконтейнером та оригінальним контейнером або є достатньо мінімальною, тобто такою, яка є “невидима” візуально для зловмисника [13]. Для математичного оцінювання виконання вимоги непомітності використовують наступні показники: максимальну та середню різницю, відношення сигнал/шум, кореляцію.

### 1.5 Методи стеганоаналізу

Основним завданням стегоаналізу є виявлення наявності прихованого повідомлення. Зазвичай, будь-які існуючі методи стегоаналізу не дають конкретної відповіді чи є вбудоване повідомлення у зображені чи немає, а лише вказують на ймовірність такого в отриманому повідомленні.

Існує два класи стегоаналізу цифрових зображень. Перший клас включає в себе направлені методи, які використовуються для знаходження вбудованого повідомлення у випадку істинного використаного стего-методу. Другий клас включає “сліпі методи”, які використовуються у випадку відсутності знань щодо попередньо використаного стеганографічного алгоритму.

Найбільш поширеними є статистичні методи стегоаналізу. До них входять наступні: метод оцінки числа переходів значень молодших біт у сусідніх елементах зображення, метод оцінки частот появи k-бітових серій у масиві НЗБ елементів контейнера, аналіз розподілу елементів зображень на площині, перевірка розподілу елементів на монотонність, аналіз гістограм, аналіз розподілу пар значень на основі критерію $\chi^2$.

Розглянемо коротко метод аналізу розподілу пар значень на основі критерію $\chi^2$. Зрозуміло, що необхідно отримати гістограму зображення додатково оцінити розподілу пар значень отриманої попередньо гістограми. Зауважмо відмінність для різних форматів даних, для прикладу, для зображення у форматі *.bmp пари значень формуються тільки на основі значень пікселів, для *jpeg – з використанням

коефіцієнтів дискретного косинусного перетворення, які відрізняються молодшим бітом [14].

В основі виявлення вбудованого повідомлення з використанням критерію $\chi^2$ є залежність між частотами появи сусідніх бітів контейнера, які є розкинуті відносно значення частоти середнього арифметичного цих елементів. У випадку вкраплення зайвої інформації частоти елементів зі значеннями $2N$ і $2N+1$ близькі за значенням або і рівними, що є рідкісним явищем при дослідженні звичайного зображення (без вкраплення, з пустим контейнером).

Тому аналіз на основі критерію $\chi^2$ полягає в тому, щоб знайти такі близькі значення та розрахувати ймовірність приховання додаткової інформації на основі оціненої подібності частот парних та непарних елементів контейнера. Даний алгоритм є послідовним та універсальний, оскільки завдяки йому можна досліджувати будь-які зображення отримані після застосування до них стеганографічних методів [14].

Зауважимо, що результати стегоаналізу із використанням методу за критерієм $\chi^2$ все таки залежать від методу приховання даних (рис. 1.8 та 1.9).

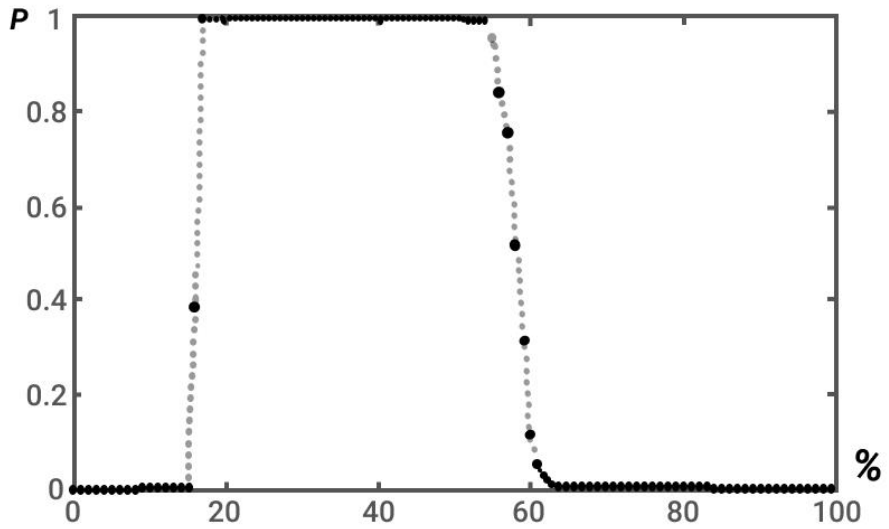



Рисунок 1.8 – Ймовірність послідовного вкраплення за критерієм $\chi^2$ при послідовній заміні НЗБ контейнера

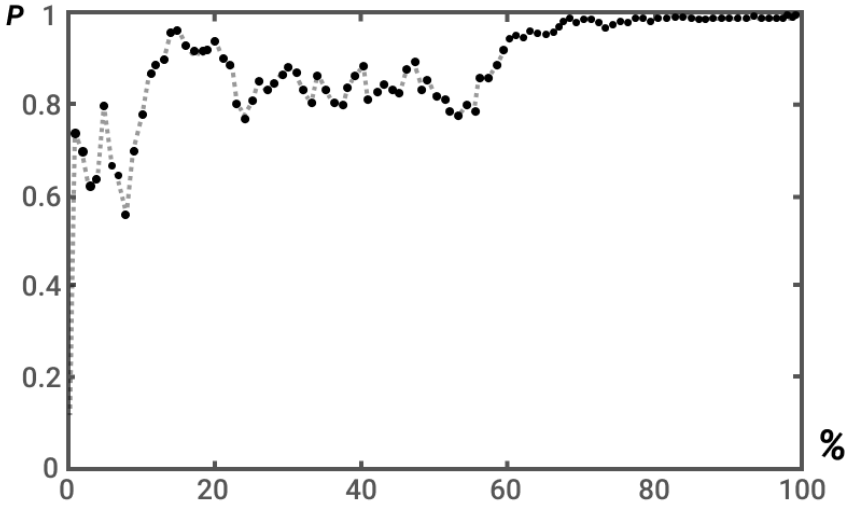



Рисунок 1.9 – Ймовірність вкраплення із заповненням за критерієм $\chi^2$ при псевдовипадковому виборі молодших бітів

Рисунки 1.8 та 1.9 демонструють відмінність результатів застосування стегоаналізу на основі критерію $\chi^2$. Як бачимо при використанні стеганографічного методу послідовної заміни НЗБ елементів контейнера і вкраплення корисного повідомлення метод повністю виявляє наявність прихованих даних (рис. 1.8), а при псевдовипадковому виборі (розподіленому вкрапленні) молодших бітів (розподіленому вкрапленні) – не виявляє.

### 1.6 Методи порівняльного аналізу та його застосування

Порівняльний аналіз являє собою загальнонауковий метод пошуку та виявлення подібності або відхилення однотипних властивостей (ознак, змін, тенденцій) об’єктів, що вивчаються, на основі зібраних статистичних даних або емпіричних досліджень.

Порівняльний аналіз використовується для вивчення об'єктів та систем технічного, органічного, символічного, соціального характеру. Головною умовою застосування методу є наявність хоча б однієї загальної властивості у досліджуваних об'єктах, в рамках якої можна розрізнити варіації досліджуваних змінних.

Існує два основних типи порівняльного аналізу:

− з’ясування істотних характеристик двох або більше пов’язаних об’єктів шляхом порівняння їх подібних властивостей;

− встановлення діаграм розвитку одного і того ж досліджуваного об'єкта шляхом порівняння його станів та властивостей у різні періоди.

Виконання порівняльного аналізу передбачає такі кроки:

− визначення об'єктів та одиниць аналізу;

− формулювання критеріїв порівняння;

− перевірка методологічної еквівалентності порівняння;

− оцінка параметрів обраних об’єктів;

− тлумачення спільних та відмінних властивостей.

Процедури порівняльного аналізу визначаються цілями конкретного дослідження. Залежно від того, як об'єкти порівнюються – у статиці чи динаміці, аналіз. Порівняльний аналіз є ефективним інструментом для перевірки гіпотез та побудови теорій, оскільки допомагає відрізнити загальні характеристики та фактори від унікальних.

Результати порівняльного аналізу є підставою для подальшого застосування алгоритму прийняття рішення, для якого перший є окремою фазою під час проектування.

### 1.7 Постановка задачі

Захист інформації від неавторизованого доступу з використанням різноманітних методів досліджувались протягом всієї історії людства. Перш за все, завжди існують умови, при яких потрібно передати секретну інформацію із певними рівнями доступу до неї. Зростання частоти появи різноманітних форматів даних також є підставою для їх організації та управління. Врахувавши зазначені фактори, дослідження у галузі стеганографії стають все більш важливими [4]. Вирішення проблеми приховування інформації також є важливим питанням у розвиненій мережевій комунікаційній інфраструктурі користувачів комп’ютерних

мереж, з розвитком якої стало можливим швидко і недорого передати необхідну інформацію адресату. Необхідно зауважити, що великі обсяги переданих документів часто супроводжуються незаконним копіюванням та порушенням авторських прав. Як результат, ці фактори є підставою для пошуку та оптимізації наявних способів приховування інформації в різноманітних форматах представлення тексту, графіки, аудіо, відео.

На сьогоднішній день для стеганографії використовується багато програмного забезпечення, яке реалізує методи інтеграції конфіденційних даних у різні типи файлів.

Метою даної кваліфікаційної роботи є дослідження стеганографічних алгоритмів приховування інформації в зображеннях, виконання порівняльного аналізу на основі вибраних критеріїв та практична реалізація вдосконаленого стеганографічного алгоритму приховування інформації в зображеннях.

Для досягнення мети запропоновано вирішити наступні завдання:

- обрання критеріїв для аналізу обраних стеганографічних алгоритмів приховування інформації в зображеннях;

- реалізація порівняльного аналізу з використанням попередньо обраних критеріїв;

- вдосконалення (модифікація) стеганографічного алгоритму прихування інформації в зображенні;

- практична реалізація вдосконаленого стеганографічного алгоритму прихування корисної інформації в зображеннях.

### 1.8 Висновки до першого розділу

За результатами огляду літературних джерел сформульовано наступні висновки:

- із збільшенням кількості різноманітних форматів файлів зростає потреба у їх безпечній передачі по каналах зв’язку, відповідно це є передумовою для розвитку стеганографічних методів;

- розглянуто сучасні стеганографічні методи приховування інформації у графічних зображеннях (просторова та частотна область), що дозволило окреслити область дослідження і зрозуміти переваги та недоліки кожного з них;

- врахувавши специфіку досліджуваної тематики було сформульовано основну мету роботи і відповідно перелік задач для її досягнення.

# 2 ПОРІВНЯЛЬНИЙ АНАЛІЗ СТЕГАНОГРАФІЧНИХ АЛГОРИТМІВ ПРИХОВУВАННЯ ІНФОРМАЦІЇ В ЗОБРАЖЕННЯХ

Метою даного розділу є детальне дослідження стеганографічних алгоритмів приховування інформації в зображеннях (просторовій та частотній області), виявлення їх переваг та недоліків, виконання порівняльного аналізу відповідних груп методів.

## 2.1 Стеганографічні методи приховування інформації в просторовій області зображення

Розглянемо детальніше кожен з методів приховування інформації в просторовій області зображення, перелік яких був наведений у розділі 1.

*Метод заміни найменш значущого біта* (НЗБ) включає в себе алгоритми, що дозволяють послідовно в кожному пікселі замінити біти його складових. Піксель можна представити у вигляді трьох чисел (модель RGB) та модифікувати кожну з цих складових так, щоб її значення майже не змінилось. Зазвичай вбудовують від одного до трьох біт повідомлення в один піксель.

На рисунку 2.1 зображено декомпозицію пікселя на складові в просторі RGB (три байти) та позначено найменш значущі біти (LSB), що будуть замінені на біти прихованого повідомлення.

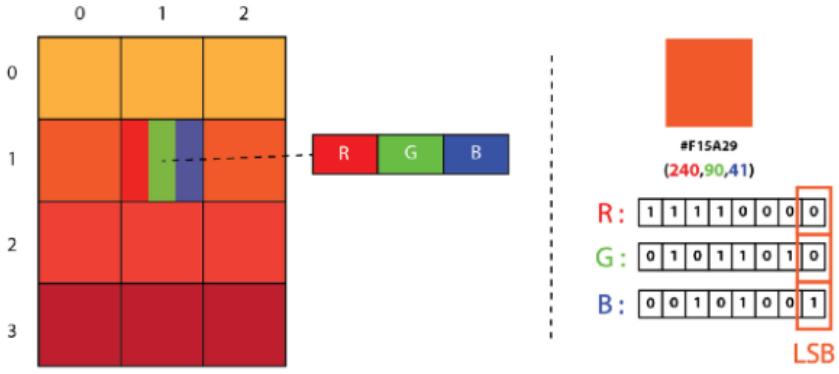



Рисунок 2.1 – Найменш значущі біти в пікселі зображення

Основною перевагою методу найменш значущого біта є те, що він дозволяє вбудовувати достатньо великі об’єми інформації в невеликі файли (пропускна спроможність прихованого каналу інформації складає від 12,5 до 30%) [3]. Недоліком НЗБ є низька стеганографічна стійкість – найменші спотворення контейнера можуть призвести до втрати повідомлення або його частини. Ще одним недоліком методу є те, що при перегляді зображення з великим масштабом чітко видно області з вбудованим повідомленням. Та все ж даний метод є достатньо популярним зважаючи на його простоту в реалізації.

*Метод псевдовипадкового інтервалу* повторює технологію вбудовування повідомлення в окремий піксель, але вибір відстані між двома такими пікселями здійснюється генератором псевдовипадкових чисел. Перевагою методу є те, що при незначних за обсягом повідомленнях відносно обсягу контейнера значно важче візуально визначити факт наявності закодованої інформації. Недолік же полягає у все ще низькій стійкості до стегоаналізу, так як тільки відстань піж вбудованими бітами може здатись випадковою.

*Метод псевдовипадкової перестановки* вдосконалює спосіб розстановки цільових пікселів тим, що їх порядок також обирається псевдовипадково. Генератор псевдовипадкових чисел (ПВЧ) будує послідовність індексів пікселів, яка надалі вказує в який піксель вбудовувати той чи інший біт секретного повідомлення. Даними алгоритмами можна забезпечити рівномірний розподіл секретних біт у просторі контейнера.

В основі *блочного методу* лежить принцип розбиття контейнера на блоки довільного розміру, в кожному з яких обраховується біт парності найменш значущих бітів. У випадку, якщо біт парності не дорівнює секретному біту, то інверсія одного з найменш значущих біт блоку змінить його значення на потрібне.

Відомо, що цей метод дуже вразливий до спотворень стеганоконтейнера. В той же момент йому властиві наступні переваги:

− вибір пікселя з блоку, в якому проводити інверсію НЗБ, можна обрати так, щоб зміни в контейнері були менш помітними;

− зменшення спотворення зображення після вбудовування можна досягти завдяки поділу зображення на більші блоки.

*Метод заміни палітри* для вбудовування повідомлення використовує порядок кольорів в палітрі пікселів зображення. Так як перестановка кольорів в палітрі не грає ролі і можна здійснити $N!$ їх перестановок, то існує можливість вбудовування невеликого повідомлення. Вагомим недоліком даного методу є те, що він вважається нестійким.

*Метод квантування зображень* будується на залежності пікселів один від одного. Вбудовування інформації проводять модифікацією різницевого сигналу $\Delta_i$. Даний метод належить до тих, яким потрібен стеганоключ. Він формується у вигляді таблиці відповідностей біт і $\Delta_i$. Якщо для поточного біта була обрахована різниця $\Delta_i$ і $b_i$ не сходиться з бітом повідомлення, то значення $\Delta_i$ потрібно замінити на таке $\Delta_j$, щоб задана умова виконувалась.

*В методі ДДКМ* (Дамстедгера-Делейгла-Квіксвотера-Мака) спочатку відбувається пошук підходящих блоків $8 \times 8$ пікселів для вбудовування. І згідно результатів виконаного пошуку відповідний біт повідомлення вбудовується в кожен піксель блоку. Метод є стійким до JPEG-компресії, так як розмір блоку відповідає алгоритмам стиснення, та має підвищену загальну стійкість за рахунок надлишковості кодування.

Основу цього методу складають процедури виокремлення в блоках зображення контрастних зон та віднесення їх пікселів до однієї із двох категорій, що отримуються у результаті накладання матриць-масок побудованих для кожного блоку за псевдовипадковим алгоритмом [15]. Даний метод потребує досить вагому кількість операцій для пошуку відповідних блоків зважаючи на те, що деяка частина зображення може не бути підходящою і як наслідок не використовуватись, та саме тому стає можливим досягнення рівноваги між якістю вбудовування повідомлення і стійкістю до модифікацій контейнера та стегоаналізу.

*Метод Куттера-Джордана-Боссена* проводить вбудовування бітів секретного повідомлення за рахунок зміни компоненти яскравості або синього кольору, що робить деформації контейнера непомітними для людського ока.

Алгоритму присутня стійкість багатьох відомих методів стегоаналізу, та видобування повідомлення може пройти невдало, тому що функції вбудовування та витягнення не є симетричними.

Зазначимо наступні та наведемо нижче відомі алгоритми, що використовують (чи модифікують) метод заміни найменш значущого біта.

Приховування всліпу (BlindHide). Приховування бітів повідомлення відбувається в пікселі один за одним зліва направо починаючи з першого рядка. Недоліком вважають випадок, коли контейнер не повністю заповнений, і відповідно верхня частина зображення буде візуально засміченою (із шумом).

Заховати-знайти (HideSeek). Порядок приховування бітів повідомлення обирається генератором псевдовипадкових чисел. Має перевагу над алгоритмом приховування всліпу тим, що контейнер буде завжди заповнений рівномірно, але наявність вбудованого повідомлення можна ідентифікувати візуально, так як не враховуються особливості контейнера.

Попередня фільтрація (FilterFirst). Суть алгоритму в тому, що біт секретного повідомлення вбудовується в такий піксель, зміна найменш значущих бітів якого буде мінімальною.

Стеганографія морського бою (BattleSteg). Серед названих алгоритмів є найскладнішим та найдосконалішим. Спочатку для біта виконується пошук найкращих для вбудовування пікселів (подібно до алгоритму попередньої фільтрації), потім з допомогою генератора псевдовипадкових чисел відбувається вибір у який піксель його вбудовувати.

## 2.2 Стеганографічні методи приховування інформації в частотній області зображення

Одним із основних методів приховування інформації в частотній області зображення є *алгоритм Коха-Жао*. Суть даного алгоритму полягає у тому, що зображення розділяється на деякі блоки $8 \times 8$ пікселів, з цих блоків виділяється матриця обраних компонент пікселів, ця матриця підлягає дискретному

косинусному перетворенню (ДКП), підміні обраних коефіцієнтів та зворотному перетворенню. Принцип вбудовування полягає у модифікації коефіцієнтів ДКП таким чином, щоб їх абсолютна різниця була більша за параметр алгоритму з знаком плюс (для вбудовування “0”) або менша за параметр алгоритму з знаком мінус (для вбудовування “1”). По суті алгоритм Коха-Жао є алгоритмом вбудовування прихованого повідомлення в процес JPEG-стиснення (рис. 2.2).

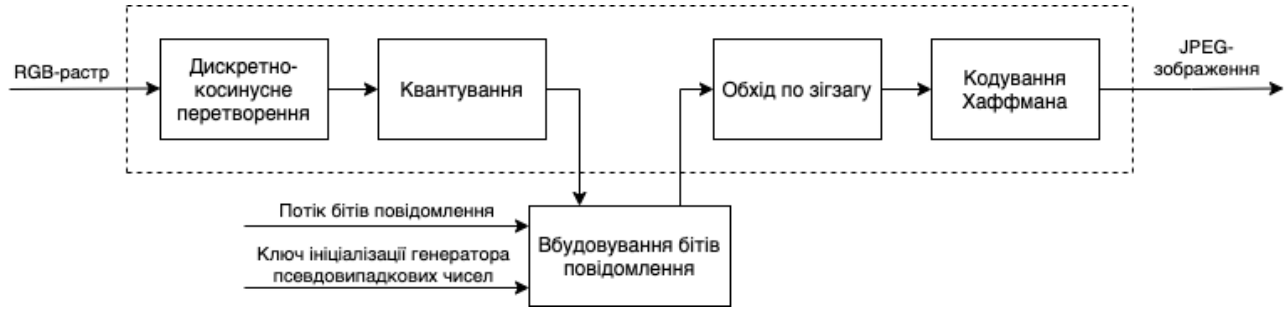



Рисунок 2.2 – Вбудовування бітів повідомлення в процесі JPEG-стиснення зображення

Основним недоліком методу Коха-Жао є висока складність обчислення ДКП. В наступних підпунктах буде розглянуто більш детальніше метод та обґрунтовано його переваги та недоліки.

*Метод Бенгама-Мемона-Ео-Юнга* оптимізує метод Коха-Жао тим, що вибір блоків відбувається таким, чином, щоб їх зміни були найменшими, і для вбудовування використовується на один коефіцієнт ДКП більше, тому зображення зазнає менше спотворень, що можна спостерігати візуально.

Під час вибору блоків для вбудовування керуються двома вимогами. У блоках мають бути відсутні різкі переходи яскравості і вони мають бути в достатній мірі монотонними.

*Метод Ху-Ву* було створено для проведення вбудовування цифрового водяного знаку (ЦВЗ) шляхом модифікації коефіцієнтів ДКП блоків обраного контейнера. Практикується створення зображення ЦВЗ чорно-білим. Перед вбудовуванням цілий масив ЦВЗ ділять на 255, а при видобуванні його перемножують на це ж число.

*При застосуванні методу Фрідріха*, зображення перетворюють в сигнал з ненульовим математичним сподіванням та визначеним стандартним відхиленням. Після виконаних маніпуляцій обраховані коефіцієнти КПД низьких частот потраплять в деяку визначену область і абсолютне значення низькочастотного коефіцієнта дискретного косинусного перетворення вхідного сигналу не буде перевищувати визначений рівень. Саме тому модифікуванню підлягають з усіх тільки низькочастотні коефіцієнти ДКП.

## 2.3 Реалізація стеганографічних алгоритмів приховування інформації в зображеннях

Для виконання порівняльного аналізу було реалізовано декілька алгоритмів на мові програмування C#. Виконане рішення складається з ядра та модульних тестів.

Ядро містить фабрику, що дозволяє конфігурувати секретні дані (клас Stego), що будуть вбудовуватись в контейнер у вигляді файлу або масиву байтів, ключ вибору порядку вбудовування пікселів для алгоритму НЗБ та константу $D$ для алгоритму Коха і Жао. Дана фабрика ще містить методи, що ініціюють вбудовування (Embed) та витягнення (Decode) повідомлення з контейнера.

Лістинг 2.1 – Фабрика Stego

```
public sealed class Stego : StegoEntry
{
    public Stego(string imagePath) : base(imagePath)
    {
    }
    public void SetSecretData(string file)
    {
        base.LoadSecretData(file);
    }
    public void SetSecretData(byte[] bytes)
    {
        base.LoadSecretData(bytes);
    }
    public Image<Rgba32> Embed(AlgorithmEnum algorithm)
    {
        if (this.image == null)
            throw new System.NullReferenceException("Image cannot be null");
```

```
        if (this.secretData == null)
            throw new System.NullReferenceException("Secret data cannot be null");
        var alg = AlgorithmFactory.Create(algorithm);
        return alg.Embed(this.image, this.secretData, this.settings);
    }
    public byte[] Decode(AlgorithmEnum algorithm)
    {
        if (this.image == null)
            throw new System.NullReferenceException("Image cannot be null");
        var alg = AlgorithmFactory.Create(algorithm);
        return alg.Decode(this.image, this.settings);
    }
    public void SetSettings(Settings settings)
    {
        this.settings = settings;
    }
}
```

В реалізації також присутня фабрика, яка в залежності від параметра, створює об’єкт відповідної реалізації алгоритму (клас AlgorithmFactory).

Лістинг 2.2 – Фабрика AlgorithmFactory

```
public static class AlgorithmFactory
{
    public static StegoAlgorithm Create(AlgorithmEnum selection)
    {
        var type = Type.GetType(typeof(StegoAlgorithm).Namespace + "." +
selection.ToString(), throwOnError: false);

        if (type == null)
        {
            throw new NullReferenceException(selection.ToString() + " is not a
known algorithm type");
        }

        if (!typeof(StegoAlgorithm).IsAssignableFrom(type))
        {
            throw new InvalidOperationException(type.Name + " does not inherit
from StegoAlghorithm");
        }

        return (StegoAlgorithm)Activator.CreateInstance(type);
    }
}
```

Лістинг 2.3 – Інтерфейс AlgorithmFactory

```
public interface IStegoAlgorithm
{
    Image<Rgba32> Embed(Image<Rgba32> baseImage, SecretData secret, Settings
settings = null);
    byte[] Decode(Image<Rgba32> stegoImage, Settings settings = null);
    int ReadSecretLength(Image<Rgba32> stegoImage, Settings settings = null);
```

```
    bool IsEmbedPossible(Image<Rgba32> image, int secretLength);
}
```

Лістинг 2.4 – Абстрактний клас StegoAlgorithm

```
public abstract class StegoAlgorithm : IStegoAlgorithm
{
    protected int SecretDataLength = 32;

    public abstract byte[] Decode(Image<Rgba32> stegoImage, Settings settings =
null);

    public abstract Image<Rgba32> Embed(Image<Rgba32> baseImage, SecretData
secret, Settings settings = null);

    public abstract bool IsEmbedPossible(Image<Rgba32> image, int secretLength);

    public abstract int ReadSecretLength(Image<Rgba32> stegoImage, Settings
settings = null);

    internal Random GetRandomGenenator(Settings settings)
    {
        return GetRandomGenenator(settings?.Key);
    }

    internal Random GetRandomGenenator(string seed)
    {
        return new Random((seed ?? string.Empty).GetHashCode());
    }
}
```

Стеганографічні алгоритми описуються інтерфейсом IStegoAlgorithm, що містить прототипи методів вбудовування та витягнення повідомлення, отримання довжини вбудованого повідомлення і перевірки можливості вбудовування повідомлення заданої довжини. Абстрактний клас StegoAlgorithm є реалізацією цього інтерфейсу та окрім абстрактних методів, описаних в інтерфейсі, містить методи отримання параметризованого генератора псевдовипадкових чисел.

### 2.3.1 Реалізація алгоритму НЗБ

Для реалізації алгоритму НЗБ був створений клас Nsb, що наслідується від абстрактного класу StegoAlgorithm.

Для функції вбудовування (додаток Г) обов’язковими параметрами є порожній контейнер, байти секретного повідомлення та секретний ключ для

генерації псевдовипадкової послідовності. Спочатку відбувається підготовка повідомлення для вбудовування. Для цього число, що означає кількість байтів повідомлення, перетворюється в масив байтів, конкатенується (об'єднання) з байтами секретного повідомлення та перетворюється в масив бітів. Наступним кроком є перевірка чи вмістить контейнер задане повідомлення. Якщо ця умова виконується, то відбувається ініціалізація генератора псевдовипадкових чисел та змінні, що знадобляться для ітерацій вбудовування повідомлення. Під час кожної ітерації генеруються випадкові координати пікселя для вбудовування поточної пари бітів. При умові, що дана точка під час сесії вбудовування обирається вперше, піксель розкладається у просторі RGB і найменш значущі біти складових R та B замінюються двома бітами повідомлення. Інакше поточна ітерація переривається. Реалізація даної функції представлена у лістингу 2.5.

Лістинг 2.5 – Реалізація алгоритму вбудовування методом НЗБ

```
public override Image<Rgba32> Embed(Image<Rgba32> baseImage, SecretData secret,
Settings settings = null)
{
    BitArray secretBits = secret.SecretWithLengthBits;
    if (IsEmbedPossible(baseImage, secretBits.Length) == false)
        throw new InvalidDataException("Secret data is to big for embending.");
    Random random = GetRandomGenenator(settings);
    int index = 0;
    while (index < secretBits.Length)
    {
        List<Tuple<int, int>> occupied = new List<Tuple<int, int>>();
        int width = random.Next(baseImage.Width);
        int height = random.Next(baseImage.Height);
        var pair = new Tuple<int, int>(width, height);
        if (occupied.Contains(pair))
        {
            continue;
        }
        occupied.Add(pair);
        var pixel = baseImage[width, height];
        pixel.R = SetLsb(pixel.R, secretBits[index]);
        pixel.B = SetLsb(pixel.B, secretBits[index + 1]);
        baseImage[width, height] = pixel;
        index += 2;
    }

    return baseImage;
}
private static byte SetLsb(byte b, bool value)
{
    return value
    // Make LSB 1
```

```
        ? (byte)(b | 1)
    // Make LSB 0
        : (byte)(b & 254);
}
```

Функція витягнення повідомлення (додаток Д) потребує заповнений контейнер (стегоконтейнер) та ключ для генерації псевдовипадкової послідовності. Спочатку відбувається ініціалізація генератора псевдовипадкових чисел. Потім ітеративно зчитуються перші вбудовані 32 біти, що містять інформацію про довжину повідомлення. У наступному кроці перевіряється коректність довжини секретного повідомлення, позитивний результат дозволяє зчитувати біти секретного повідомлення. Ітерація зчитування пари бітів починається з генерації випадкових координат пікселя. Якщо цей піксель ще не був прочитаний, то його розкладають у просторі RGB та запам’ятовують найменш значущі біти складових R і B. Реалізація цієї функції представлена у лістингу 2.6.

Лістинг 2.6 – Реалізація алгоритму витягнення методом НЗБ

```
public override byte[] Decode(Image<Rgba32> stegoImage, Settings settings = null)
{
    var random = GetRandomGenenator(settings?.Key);
    var secretBytesLengthBits = new BitArray(this.SecretDataLength);
    var index = 0;
    var occupied = new List<Tuple<int, int>>();
    while (index < this.SecretDataLength)
    {
        var width = random.Next(stegoImage.Width);
        var height = random.Next(stegoImage.Height);
        var pair = new Tuple<int, int>(width, height);
        if (occupied.Contains(pair))
        {
            continue;
        }
        occupied.Add(pair);
        var pixel = stegoImage[width, height];
        var bitR = GetLsb(pixel.R);
        secretBytesLengthBits.Set(index, bitR);
        var bitB = GetLsb(pixel.B);
        secretBytesLengthBits.Set(index + 1, bitB);
        index += 2;
    }
    var secretBytesLength =
BitConverter.ToInt32(secretBytesLengthBits.ToByteArray(), 0);
    if (secretBytesLength <= 0 || !IsEmbedPossible(stegoImage, secretBytesLength))
    {
        throw new DecodeException($"Cannot read secret from this image file.");
    }
    var secretBits = new BitArray(secretBytesLength * 8);
```

```
    while (index < secretBytesLength * 8 + this.SecretDataLength)
    {
        var width = random.Next(stegoImage.Width);
        var height = random.Next(stegoImage.Height);
        var pair = new Tuple<int, int>(width, height);
        if (occupied.Contains(pair))
        {
            continue;
        }
        occupied.Add(pair);
        var pixel = stegoImage[width, height];
        var bitR = GetLsb(pixel.R);
        secretBits.Set(index - this.SecretDataLength, bitR);
        var bitB = GetLsb(pixel.B);
        secretBits.Set(index - this.SecretDataLength + 1, bitB);
        index += 2;
    }
    var secretBytes = secretBits.ToByteArray();
    return secretBytes;
}

private static bool GetLsb(byte b)
{
    return (b & 1) != 0;
}
```

Представимо результати виконання алгоритму заміни НЗБ. Для цього було обрано зображення-оригінал, тобто по суті зображення із незаповненим стегоконтейнером (рис. 2.3).

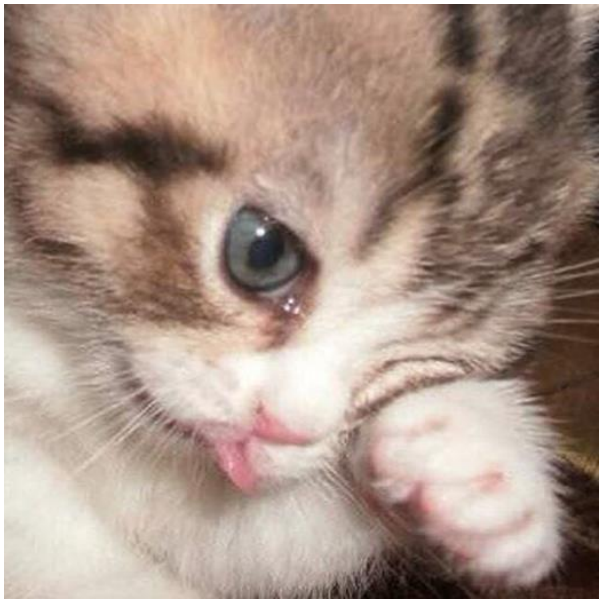


Рисунок 2.3 – Незаповнений контейнер

На рис. 2.4 зображено зображення уже із вбудованим повідомленням методом заміни НЗБ.

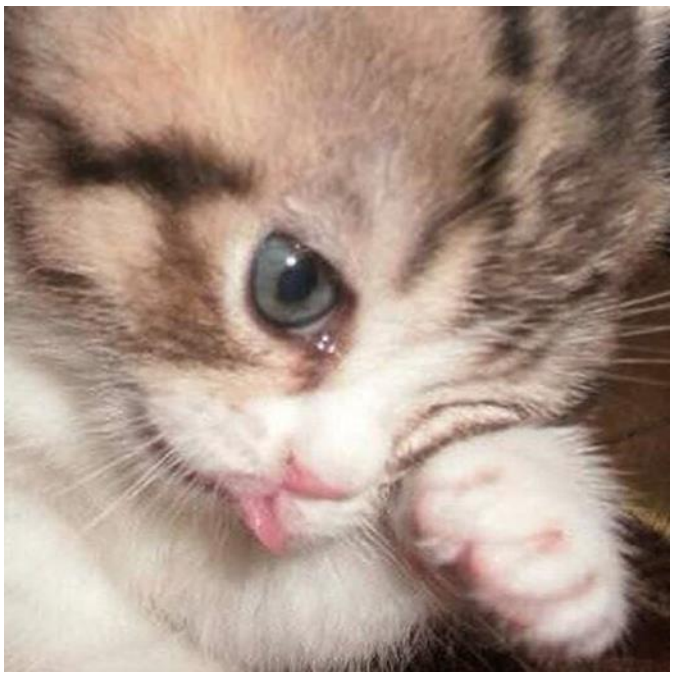


Рисунок 2.4 – Заповнений контейнер алгоритмом НЗБ

Порівнявши рисунки 2.3 і 2.4, дослідивши максимальну пропуску здатність передачі повідомлення з використанням такого типу зображення, робимо висновок, що при довжині повідомлення в 40000 байтів (приблизно 20% від максимальної пропускної здатності) візуально неможливо відрізнити порожній контейнер від заповненого, отриманого з використанням алгоритму НЗБ.

### 2.3.2 Реалізація алгоритму Коха і Жао

Для реалізації алгоритму Коха і Жао був створений клас ZhaoKoch, що наслідується від абстрактного класу StegoAlgorithm.

Для функції вбудовування (додаток Ж) обов’язковими параметрами є порожній контейнер, байти секретного повідомлення та параметр D. Крок підготовки секретного повідомлення такий же як і в алгоритмі НЗБ. Далі виконується перевірка чи повідомлення поміститься в наданий контейнер. При умові, що контейнер достатньо великий для повідомлення, в кожен блок $8 \times 8$ пікселів один за одним вбудовується один секретний біт. Ітерація починається з розкладання кожного пікселя блоку в просторі YCbCr та формування матриці компонент Y. Наступним кроком є виконання операції дискретного косинусного

перетворення отриманої матриці. З матриці коефіцієнтів обираються два та модифікуються так, щоб їх абсолютна різниця була більшою за D (біт “0”) або меншою за -D (біт “1”). Після описаних маніпуляцій відбувається зворотне дискретне косинусне перетворення і модифікована матриця компонент Y вбудовується назад в блок пікселів. Реалізація цього алгоритму представлена у лістингу 2.7.

Лістинг 2.7 – Реалізація алгоритму вбудовування методом Коха і Жао

```
public override Image<Rgba32> Embed(Image<Rgba32> baseImage, SecretData secret,
Settings settings = null)
{
    var secretBits = secret.SecretWithLengthBits;
    if ((baseImage.Width / 8) * (baseImage.Height / 8) < secretBits.Length)
    {
        throw new InvalidDataException("Secret data is to big for embending.");
    }
    var d = settings?.D ?? 5;
    int width = 0;
    int height = 0;
    int index = 0;
    while (index < secretBits.Length)
    {
        if (width + 8 > baseImage.Width)
        {
            height += 8;
            width = 0;
        }
        if (height + 8 >= baseImage.Height)
        {
            break;
        }
        var luminanceMatrix = GetLuminanceMatrix(baseImage, width, height);
        var yMatrix = luminanceMatrix.GetY();
        var matrixWithBit = InsertOneBit(yMatrix, secretBits.Get(index), d);
        luminanceMatrix = luminanceMatrix.SetY(matrixWithBit);
        SetLuminance(baseImage, luminanceMatrix, width, height);
        index++;
        width += 8;
    }
    return baseImage;
}

public float[][] InsertOneBit(float[][] matrix, bool bit, float d)
{
    float[][] quantizeMatrix = (Dct(matrix));
    float k1 = quantizeMatrix[3][4];
    float k2 = quantizeMatrix[4][3];
    if (bit)
    {
        while (!(Math.Abs(k1) - Math.Abs(k2) < -(d + 5)))
            k2 += (k2 < 0 ? -1 : 1);
    }
    else
```

```
    {
        while (!(Math.Abs(k1) - Math.Abs(k2) > (d + 5)))
            k1 += (k1 < 0 ? -1 : 1);
    }
    quantizeMatrix[3][4] = k1;
    quantizeMatrix[4][3] = k2;
    float[][] outMatrix = DctInv((quantizeMatrix));
    return outMatrix;
}
```

Функція отримання вбудованого повідомлення (додаток И) починається із зчитування перших 32 блоків стегоконтейнера. З отриманих бітів формується інформація про кількість блоків, які потрібно зчитати після цього. Після цього, якщо зчитана довжина повідомлення була правильною, продовжується послідовне зчитування всіх заповнених блоків контейнера. Ітерація читання блоку починається з формування матриці компонент Y в просторі YCbCr. Для отриманої матриці проводять дискретне косинусне перетворення і якщо абсолютна різниця обраних коефіцієнтів більша за D, то зчитаний біт має значення “0”, інакше “1”. Реалізація цієї функції представлена у лістингу 2.8.

Лістинг 2.8 – Реалізація алгоритму витягнення методом Коха і Жао

```
public override byte[] Decode(Image<Rgba32> stegoImage, Settings settings = null)
{
    var d = settings?.D ?? 5;
    var secretLengthBits = new BitArray(32);
    int width = 0;
    int height = 0;
    int index = 0;
    while (index < 32)
    {
        if (width + 8 > stegoImage.Width)
        {
            height += 8;
            width = 0;
        }
        if (height + 8 >= stegoImage.Height)
        {
            break;
        }
        var luminanceMatrix = GetLuminanceMatrix(stegoImage, width, height);
        var yMatrix = luminanceMatrix.GetY();
        var bit = ReadOneBit(yMatrix, d);
        secretLengthBits.Set(index, bit);
        width += 8;
        index += 1;
    }
    var secretLength = BitConverter.ToInt32(secretLengthBits.ToByteArray(), 0);
    if (secretLength <= 0 || (stegoImage.Width / 8) * (stegoImage.Height / 8) <
```

```
secretLength)
    {
        throw new DecodeException($"Cannot read secret from this image file.");
    }
    var secretBits = new BitArray(secretLength * 8);
    while (index < secretLength * 8 + 32)
    {
        if (width + 8 > stegoImage.Width)
        {
            height += 8;
            width = 0;
        }
        if (height + 8 >= stegoImage.Height)
        {
            break;
        }
        var luminanceMatrix = GetLuminanceMatrix(stegoImage, width, height);
        var yMatrix = luminanceMatrix.GetY();
        var bit = ReadOneBit(yMatrix, d);
        secretBits.Set(index - 32, bit);
        width += 8;
        index += 1;
    }
    return secretBits.ToByteArray();
}

private bool ReadOneBit(float[][] matrix, float d)
{
    float[][] quantizeMatrix = Dct(matrix);
    float k1 = quantizeMatrix[3][4];
    float k2 = quantizeMatrix[4][3];
    return Math.Abs(k1) - Math.Abs(k2) < -d;
}
```

Представимо результати виконання стеганографічного методу Коха-Жао (рис. 2.5) на основі того ж самого зображення-оригіналу (рис. 2.3).

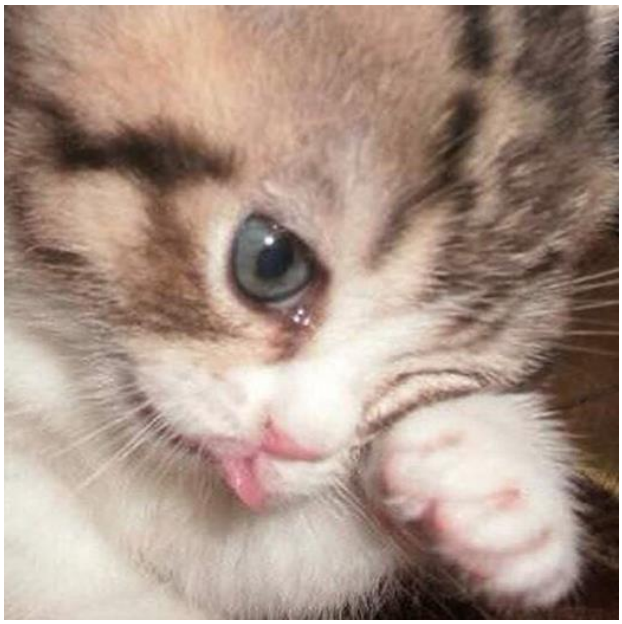


Рисунок 2.5 – Заповнений контейнер алгоритмом Коха і Жао

Виконаємо порівняння рисунків 2.3 та 2.5, робимо наступний висновок: заповнений контейнер алгоритмом Коха-Жао повідомленням з довжиною 160 байтів (приблизно 20% від максимальної ємності для даного алгоритму) візуально ідентичний порожньому контейнеру, тобто візуально розрізнити зоровою системою людиною практично неможливо.

### 2.4 Критерії оцінювання стегосистем

При виконанні будь-якого типу аналізу надзвичайно важливо обрати такі критерії, щоб можна було однозначно виконати порівняння та прийняти відповідне рішення. З цією метою для порівняння методів стеганографії можна було обрано наступні критерії оцінки стегосистем.

Існують *кількісні* критерії, які обчислюються на основі значень пікселі зображення, що дозволяє їм здійснювати оцінку ефективності стеганографічних методів [3]. Наведемо декілька з них, які використаємо для виконання порівняльного аналізу для досягнення поставленої задачі.

MD (Maximum Difference) – максимальна різниця. Описує різницю між порожнім і заповненим контейнером. Незначне значення MD вказує на високу якість зображення.

AD (Absolute Difference) – абсолютна різниця. Описує середню абсолютну різницю між порожнім і заповненим контейнером. Мале значення AD вказує на високу якість зображення [8].

IF (Image Fidelity) – якість зображення. Описує на скільки порожній контейнер відповідає заповненому. Даний критерій дозволяє оцінити успішність візуальної атаки, так як даний тип атак характеризується здатністю людини виявляти відмінності у зображеннях [6].

SNR (Signal Noise Ratio) – співвідношення сигналу до шуму. Даний параметр обернено пропорційний до величини спотворення контейнера.

NC (Normalized Cross-Correlation) – нормована взаємна кореляція. Описує зв'язок між не заповненим та заповненим контейнером. Чим ближче значення цього критерію до 1, тим подібніші вихідне та модифіковане зображення.

Можна виділити наступні якісні критерії, за допомогою яких можна характеризувати стеганосистеми:

Пропускна здатність (Capacity) – характеризує об'єм секретних даних, що можна вбудувати в контейнер за допомогою визначеного методу [8].

Стійкість (Robustness) – характеризує спроможність витягнути вбудовані секретні дані з контейнера, який був попередньо модифікований додаванням різноманітних шумів, стисненням, масштабуванням, зміною контрасту тощо. Також варто зазначити, що стійкість не передбачає захист від атак, що базуються на основі знань про використані алгоритми.

Невидимість (Invisibility) – описує неможливість відрізнити заповнений контейнер від пустого візуально, тобто за допомогою зорової системи людини. Можна назвати вбудовані дані невидимими, при умові, що середньостатистична людина не здатна підтвердити наявність закодованого повідомлення [3][8].

Іншим чином такі поняття можна описати за допомогою кількісного показника IF, що вказує на відсоток відповідності вихідного зображення та модифікованого.

Захищеність (Security) – характеризує нездатність цілеспрямованими атаками отримати інформацію з контейнера, якщо є відомими алгоритми вбудовування та витягнення, або провести часткову її модифікацію [3][6-8].

Складність вбудовування/вилучення – вказує на час вбудовування та витягнення повідомлення з контейнера шляхом кількості елементарних операцій даних процедур.

Вище представлені характеристики не можуть одночасно описувати усі стеганосистеми з позитивної сторони, так як їм це не дозволяє взаємо виключність.

Наприклад, якщо існує вимога високої стійкості системи до стегоаналізу, то його пропускна здатність не може бути великою. Також при умові, що

повідомлення повинно бути досить об’ємним, то не можливо отримати надзвичайно високу стійкість до стегоаналізу[3][8].

## 2.5 Порівняльний аналіз стеганографічних алгоритмів приховування інформації в зображеннях

В процесі виконання роботи було реалізовано 5 алгоритмів в просторовій області, а саме: НЗБ, псевдовипадкового інтервалу, псевдовипадкової перестановки, Куттера-Джордана-Боссена, квантування зображення, та 3 алгоритми в частотній області: Коха і Жао, Ху і Ву, Фрідріх. Кожен алгоритм був виконаний на множині тренувальних зображень, який в подальшому називатимемо набором зображень-оригіналів, обсяг якого становив 1000 елементів. Для виконання стеганогографічних алгоритмів було використано контейнер з наповненістю в 20%.

В результатів було отримано статистики кількісних критеріїв оцінювання стеганосистем, а їх усереднені значення представлені у вигляді таблиці 2.1.

Таблиця 2.1 – Усереднені кількісні критерії оцінювання стеганосистем

| Область вбудовування | Алгоритм | Критерій спотворення зображення | | | | |
|---|---|---|---|---|---|---|
| | | MD | AD | IF | SNR | NC |
| Просторова область | НЗБ | 1 | 0.512391 | 0.999850 | 52134 | 0.999518 |
| | Псевдовипадкового інтервалу | 1 | $7.53*10^{-3}$ | ~1 | $3.047*10^{6}$ | 0.999993 |
| | Псевдовипадкової перестановки | 1 | $6.03*10^{-3}$ | ~1 | $4.721*10^{6}$ | 0.999997 |
| | Куттера-Джордана-Боссена | 35 | 241.167 | 0.995126 | 193.658 | 0.989121 |
| | Квантування зображення | 4 | $8.95*10^{-3}$ | ~1 | $2.443*10^{6}$ | ~1 |
| Частотна область | Коха-Жао | 43 | 130.057 | 0.993027 | 198.461 | 0.985396 |
| | Ху-Ву | 86 | 0.793275 | 0.966548 | 28.534 | 0.861655 |
| | Фрідріх | 91 | 709.529 | 0.966703 | 31.382 | 0.914673 |

В процесі виконання вище перелічених алгоритмів було зауважено наступні недоліки при реалізації та отримання вбудованого повідомлення, які наведено в таблиці 2.2.

Таблиця 2.2 – Недоліки алгоритмів вбудовування інформації в зображення

| Область вбудовування | Алгоритм | Недоліки |
|---|---|---|
| Просторова область | НЗБ | Висока вразливість до модифікацій контейнера. Неоднозначність декодування. |
| | Псевдовипадкового інтервалу | Висока вразливість до модифікацій контейнера. Неоднозначність декодування. |
| | Псевдовипадкової перестановки | Висока вразливість до модифікацій контейнера. Відносно велика кількість операцій для вбудовування і витягнення. |
| | Куттера-Джордана-Боссена | Висока вразливість до модифікацій контейнера. Відносно сильно спотворює зображення. Низька стійкість до стегоаналізу. |
| | Квантування зображення | Висока вразливість до модифікацій контейнера. Неоднозначність декодування. |
| Частотна область | Коха і Жао | Відносно велика кількість операцій для вбудовування і витягнення. Низька стійкість атак до сліпого перебору. |
| | Ху і Ву | Мала ємність контейнера. Низька стійкість до статистичних тестів. |
| | Фрідріх | Відносно велика кількість операцій для вбудовування і витягнення. Мала ємність контейнера. |

Для виконання порівняльного аналізу стеганографічних алгоритмів приховування інформації в зображеннях на основі результатів представлених у таблиці 2.1 та прийняття рішення щодо пошуку оптимального алгоритму було обрано метод аналізу ієрархій (MAI).

MAI є надзвичайно простим у виконанні та інтуїтивно зрозумілим. Розроблений американським вченим Томасом Сааті у 1970-х роках, який активно використовувався ним при прийнятті складних управлінських рішень в сферах управління бізнесом та розробки продуктів. Метод не дає однозначної відповіді щодо найкращого та оптимального варіанту, а дозволяє на основі прорангованих критеріїв ітеративно визначити його.

Розглянемо детально алгоритм MAI на основі конкретного випадку, а саме на результатах оцінених кількісних критерії щодо роботи стеганосистеми (табл. 2.1). Метод має обов'язкові три складові:

− кінцеву мету або проблему, яку потрібно вирішити;

− всі можливі рішення та альтернативи;

− критерії, згідно яких будемо оцінювати відповідність альтернатив щодо вирішення проблеми.

Основними етапами MAI, що потрібно виконати для його реалізації, є наступні:

− структуризація проблеми у вигляді ієрархічної структури із кінцевою метою на верхньому рівні, критеріїв на другому рівні, альтернатив на третьому рівні;

− отримання матриці попарних порівнянь, що визначатиме відносну важливість різних критеріїв щодо досягнення поставленої мети;

− заповнення матриці попарних порівнянь альтернатив для кожного з критеріїв;

− нормалізація значень у всіх матрицях;

− обчислення коефіцієнтів важливості для критерії та альтернатив відповідно;

− оцінювання узгодженості матриці;

− обчислення загального критерію якості кожної альтернативи з використанням формули:

$$Q^{gl}(a_j) = \sum_{i=1}^{N} w_i V_{ij}, \tag{2.1}$$

де $Q^{gl}(a_j)$ – глобальний якісний критерій для альтернативи $a_j$;

$w_i$ – коефіцієнт важливості критерію $Q_i$;

$V_{ij}$ – коефіцієнт важливості альтернативи $a_j$ за критерієм $Q_i$.

Для отримання матриці попарних порівнянь, що визначатиме відносну важливість різних критеріїв щодо вказаної мети використовують шкалу Сааті, що представлена у вигляді таблиці 2.3.

Таблиця 2.3 – Шкала значень важливості Сааті

| Значення важливості | Стандартне значення | Обернене значення |
|---|---|---|
| Однакова важливість | 1 | 1 |
| Незначна перевага | 3 | 1/3 |
| Значна перевага | 5 | 1/5 |
| Дуже значна перевага | 7 | 1/7 |
| Абсолютна перевага | 9 | 1/9 |
| Посередні значення | 2, 4, 6, 8 | 1/2, 1/4, 1/6, 1/8 |

Розглянемо конкретний випадок, а саме пошук оптимального алгоритму приховування інформації для початку просторовій області графічного зображення, що становить мету порівняльного аналізу.

Маємо наступні критерії: MD, AD, SNR, NC.

Перелік альтернатив, що відображають досліджувані методи приховування інформації у просторовій області: НЗБ ($a_1$), псевдовипадкового інтервалу ($a_2$), псевдовипадкової перестановки ($a_3$), Куттера-Джордана-Боссена ($a_4$), квантування зображення ($a_5$).

Наведемо матрицю попарних порівнянь, що задаватиме відносну важливість визначених критеріїв щодо обрання оптимального методу, отриману на основі використання таблиці 2.3 та представимо у вигляді таблиці 2.4.

Таблиця 2.4 – Матриця попарних порівнянь визначних критеріїв

| | $Q_1$, MD | $Q_2$, AD | $Q_3$, IF | $Q_4$, SNR | $Q_5$, NC | Нормалізовані власні вектори |
|---|---|---|---|---|---|---|
| $Q_1$, MD | 1 | 1/3 | 9 | 1/7 | 3 | 0,1857 |
| $Q_2$, AD | 3 | 1 | 7 | 1/5 | 3 | 0,2707 |
| $Q_3$, IF | 1/9 | 1/7 | 1 | 1/3 | 1/5 | 0,0448 |
| $Q_4$, SNR | 1/7 | 5 | 3 | 1 | 3 | 0,3908 |
| $Q_5$, NC | 1/3 | 1/3 | 5 | 1/3 | 1 | 0,1077 |
| Сума | | | | | | 1 |

Наступним кроком є побудова попарного порівняння значень критерію щодо кожної із попередньо заданих альтернатив. Представимо ці результати у вигляді таблиць 2.5-2.9.

Таблиця 2.5 – Матриця попарних порівнянь альтернатив щодо критерію $Q_1$, MD

| $Q_1$, MD | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | Нормалізовані власні вектори |
|---|---|---|---|---|---|---|
| $a_1$ | 1 | 2 | 2 | 9 | 3 | $V_{11} = 0{,}3690$ |
| $a_2$ | 1/2 | 1 | 2 | 9 | 3 | $V_{21} = 0{,}2800$ |
| $a_3$ | 1/2 | 1/2 | 1 | 9 | 3 | $V_{31} = 0{,}2125$ |
| $a_4$ | 1/9 | 1/9 | 1/9 | 1 | 1/7 | $V_{41} = 0{,}0264$ |
| $a_5$ | 1/3 | 1/3 | 1/3 | 7 | 1 | $V_{51} = 0{,}1118$ |

Таблиця 2.6 – Матриця попарних порівнянь альтернатив щодо критерію $Q_2$, AD

| $Q_2$, AD | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | Нормалізовані власні вектори |
|---|---|---|---|---|---|---|
| $a_1$ | 1 | 1/5 | 1/3 | 9 | 1/5 | $V_{12} = 0{,}0968$ |
| $a_2$ | 5 | 1 | 1/2 | 9 | 2 | $V_{22} = 0{,}2885$ |
| $a_3$ | 3 | 2 | 1 | 9 | 3 | $V_{32} = 0{,}3779$ |
| $a_4$ | 1/9 | 1/9 | 1/9 | 1 | 1/9 | $V_{42} = 0{,}0238$ |
| $a_5$ | 5 | 1/2 | 1/3 | 9 | 1 | $V_{52} = 0{,}2128$ |

Таблиця 2.7 – Матриця попарних порівнянь альтернатив щодо критерію $Q_3$, IF

| $Q_3$, IF | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | Нормалізовані власні вектори |
|---|---|---|---|---|---|---|
| $a_1$ | 1 | 1/2 | 1/2 | 1 | 1/2 | $V_{13} = 0{,}1303$ |
| $a_2$ | 2 | 1 | 1 | 2 | 1 | $V_{23} = 0{,}2606$ |
| $a_3$ | 2 | 1 | 1 | 2 | 1 | $V_{33} = 0{,}2606$ |
| $a_4$ | 1 | 1/2 | 1/2 | 1 | 2 | $V_{43} = 0{,}1804$ |
| $a_5$ | 1 | 1 | 1 | 1/2 | 1 | $V_{53} = 0{,}1680$ |

Таблиця 2.8 – Матриця попарних порівнянь альтернатив щодо критерію $Q_4$, SNR

| $Q_4$, SNR | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | Нормалізовані власні вектори |
|---|---|---|---|---|---|---|
| $a_1$ | 1 | 1/5 | 1/7 | 5 | 1/5 | $V_{14} = 0{,}0699$ |
| $a_2$ | 5 | 1 | 3 | 7 | 3 | $V_{24} = 0{,}4289$ |
| $a_3$ | 7 | 1/3 | 1 | 7 | 3 | $V_{34} = 0{,}2966$ |
| $a_4$ | 1/5 | 1/7 | 1/7 | 1 | 1/7 | $V_{44} = 0{,}0309$ |
| $a_5$ | 5 | 1/3 | 1/3 | 7 | 1 | $V_{54} = 0{,}1744$ |

Таблиця 2.9 – Матриця попарних порівнянь альтернатив щодо критерію $Q_5$, NC

| $Q_5$, NC | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | Нормалізовані власні вектори |
|---|---|---|---|---|---|---|
| $a_1$ | 1 | 1/2 | 1 | 2 | 1/2 | $V_{15} = 0{,}1646$ |
| $a_2$ | 2 | 1 | 1 | 2 | 1/3 | $V_{25} = 0{,}0206$ |
| $a_3$ | 1 | 1 | 1 | 2 | 1/2 | $V_{35} = 0{,}1849$ |
| $a_4$ | 1/2 | 1/2 | 1/2 | 1 | 1/2 | $V_{45} = 0{,}1089$ |
| $a_5$ | 1 | 3 | 2 | 2 | 1 | $V_{55} = 0{,}3352$ |

Останнім кроком перед прийняттям рішення стосовно найкращої альтернативи щодо методу приховування інформації в просторовій області зображення є обчислення значення глобального критерію з використанням формули (2.1):

$$Q^{gl}(a_1) = \sum_{i=1}^{5} w_i V_{1j} = 0{,}1456$$

$$Q^{gl}(a_2) = \sum_{i=1}^{5} w_i V_{2j} = 0{,}3116$$

$$Q^{gl}(a_3) = \sum_{i=1}^{5} w_i V_{3j} = 0{,}2892$$

$$Q^{gl}(a_4) = \sum_{i=1}^{5} w_i V_{4j} = 0{,}0432$$

$$Q^{gl}(a_5) = \sum_{i=1}^{5} w_i V_{5j} = 0{,}1901$$

В результаті застосуванні MAI було обрано оптимальний метод приховування інформації у просторовій області зображення, а саме: метод псевдовипадкової перестановки.

Виконаємо аналогічно MAI і для методів приховування інформації у частотній області графічного зображення.

Вважаємо критерії аналогічними до попереднього прикладу, тому будемо використовувати матрицю попарних порівнянь важливості критерію представлену таблицею 2.4.

Перелік альтернатив, що відображають досліджувані методи приховування інформації у частотній області: метод Коха-Жао ($a_1$), Ху-Ву ($a_2$), Фрідріха ($a_3$).

Наступним кроком є побудова попарного порівняння значень критерію щодо кожної із попередньо заданих альтернатив. Представимо ці результати у вигляді таблиць 2.10-2.14.

Таблиця 2.10 – Матриця попарних порівнянь альтернатив (частотна область) щодо критерію $Q_1$, MD

| $Q_1$, AD | $a_1$ | $a_2$ | $a_3$ | Нормалізовані власні вектори |
|---|---|---|---|---|
| $a_1$ | 1 | 7 | 9 | $V_{11} = 0{,}7853$ |
| $a_2$ | 1/7 | 1 | 3 | $V_{21} = 0{,}1488$ |
| $a_3$ | 1/9 | 1/3 | 1 | $V_{31} = 0{,}0657$ |

Таблиця 2.11 – Матриця попарних порівнянь альтернатив (частотна область) щодо критерію $Q_2$, AD

| $Q_2$, MD | $a_1$ | $a_2$ | $a_3$ | Нормалізовані власні вектори |
|---|---|---|---|---|
| $a_1$ | 1 | 2 | 9 | $V_{12} = 0{,}6153$ |
| $a_2$ | 1/2 | 1 | 5 | $V_{22} = 0{,}3186$ |
| $a_3$ | 1/9 | 1/5 | 1 | $V_{32} = 0{,}0660$ |

Таблиця 2.12 – Матриця попарних порівнянь альтернатив (частотна область) щодо критерію $Q_3$, IF

| $Q_3$, IF | $a_1$ | $a_2$ | $a_3$ | Нормалізовані власні вектори |
|---|---|---|---|---|
| $a_1$ | 1 | 3 | 3 | $V_{13} = 0{,}6$ |
| $a_2$ | 1/3 | 1 | 1 | $V_{23} = 0{,}2$ |
| $a_3$ | 1/3 | 1 | 1 | $V_{33} = 0{,}2$ |

Таблиця 2.13 – Матриця попарних порівнянь альтернатив (частотна область) щодо критерію $Q_4$, SNR

| $Q_4$, SNR | $a_1$ | $a_2$ | $a_3$ | Нормалізовані власні вектори |
|---|---|---|---|---|
| $a_1$ | 1 | 9 | 9 | $V_{14} = 0{,}8082$ |
| $a_2$ | 1/9 | 1 | 1/3 | $V_{24} = 0{,}0622$ |
| $a_3$ | 1/9 | 3 | 1 | $V_{34} = 0{,}1295$ |

Таблиця 2.14 – Матриця попарних порівнянь альтернатив щодо критерію $Q_5$, NC

| $Q_5$, NC | $a_1$ | $a_2$ | $a_3$ | Нормалізовані власні вектори |
|---|---|---|---|---|
| $a_1$ | 1 | 7 | 9 | $V_{15} = 0{,}7791$ |
| $a_2$ | 1/7 | 1 | 1/5 | $V_{25} = 0{,}0598$ |
| $a_3$ | 1/9 | 5 | 1 | $V_{35} = 0{,}1609$ |

$$Q^{gl}(a_1) = \sum_{i=1}^{5} w_i V_{1j} = 0{,}7390$$

$$Q^{gl}(a_2) = \sum_{i=1}^{5} w_i V_{2j} = 0{,}1535$$

$$Q^{gl}(a_3) = \sum_{i=1}^{5} w_i V_{3j} = 0{,}1069$$

В результаті застосуванні MAI було обрано оптимальний метод приховування інформації у частотній області графічного зображення, а саме: метод Коха-Жао.

## 2.6 Висновки до другого розділу

Основними результатами роботи, представлених у другому розділі роботи вважаємо наступні:

- виконано практичну реалізацію стеганографічних методів із використанням мови програмування C#, фреймворку .NET Core 2.2, та бібліотеки обробки зображень SixLabors, що дозволило оцінити критерії оцінювання стегосистеми для кожного досліджуваного методу;

- виконано детально порівняльний аналіз в умовах багатокритеріальності з використанням методу аналізу ієрархій, зважаючи на його простоту та зрозмулість у реалізації;

- отримано та обрано оптимальні методи приховування інформації у графічних зображеннях, а саме: у просторовій області – метод псевдовипадкової перестановки, у частотній – метод Коха-Жао, які будуть використані у подальшій модифікації.

# 3 РОЗРОБКА КОМБІНОВАНОГО АЛГОРИТМУ НА ОСНОВІ ПРОАНАЛІЗОВАНИХ

Метою даного розділу є ідентифікація недоліків в реалізованих у розділі 2 стеганографічних алгоритмах приховування інформації в зображеннях, та пошук можливих рішень з метою зменшення впливу недоліків на якість стегосистеми.

## 3.1 Ідентифікація недоліків стеганографічних алгоритмів приховування інформації в зображеннях

Завдання стеганографії вимагають певного співвідношення між стійкістю вбудованих даних у контейнер до непередбачуваних факторів і їх об’ємом. Для існуючих алгоритмів, що призначені для приховання інформації у зображеннях, має місце обернено пропорційна залежність надійності системи від об’єму вбудовуваних даних, яка говорить про те, що збільшення останнього істотно знижує надійність системи.

Тому існує перспектива прийняття оптимального рішення при виборі між кількістю приховуваних даних і рівнем стійкості до можливої модифікації чи стегоаналізу. Використовуючи будь-який метод, завдяки надлишковості інформації існує ймовірність підвищити ступінь надійності приховання, знижуючи при цьому пропускну здатність (об’ємом приховуваних даних).

В той же час невеликі зміни у контейнері можуть пошкодити вбудоване повідомлення, а якщо пошкодиться інформація про довжину повідомлення, то витягнення навіть частини інформації буде проблематичним.

Алгоритми приховування в частотній області зображення зазвичай є більш стійкими, ніж в просторовій, до коригування кольорів, яскравості, контрасту або гами. Зазначимо, що методи приховування в частотній області є вразливими до таких модифікацій як вибіркове накладання спотворень, перезаписання пікселів пензлем в графічному редакторі.

Отож, основним недоліком при використанні методів приховування інформації в зображеннях є зменшення надійності стегосистеми при істотному збільшенні обсягу корисної інформації (вбудованого повідомлення).

### 3.2 Підвищення надійності стеганографічних алгоритмів приховування інформації в зображеннях

Для вирішення проблеми втрати частини повідомлення в стегоконтейнері внаслідок його деформацій було вирішено збільшити стійкість за рахунок надлишковості вбудовуваних даних. Інструментом вирішення цього завдання можна обрати завадостійке кодування, так як це один з найефективніших способів забезпечення високої стійкості до втрат при зберіганні та передачі інформації.

Завадостійкі коди (коригувальні коди) використовують для відновлення інформації, пошкодженої після передачі по каналах даних або невдалого збереження на носії. Кількість перевірочних бітів у заводостійких кодах залежить від кількості помилок, що повинні бути виправлені завдяки обраному алгоритму кодування. Для прикладу, якщо кількість інформаційних бітів становить 16, а кількість помилок для виправлення 4, то кількість перевірочних бітів дорівнюватиме 9. Загальна довжина кодової комбінації становитиме 25 бітів, надмірність буде більшою за 0,5. Внаслідок цього, потрібно звернути увагу на збільшення кількості перевірочних бітів із приростом кількості помилок, що потребують виправлення. Необхідно звернути увагу, що об’єктом дослідження є процес приховування інформації саме у зображеннях, а збільшення обсягу приховуваної інформації вплине на якість вихідного зображення і буде відчутно для сприйняття.

Для обрання завадостійкого алгоритму кодування розглянемо лінійку кодів, що здатні виправляти одну помилку. До даної групи належать коди з кодовою відстанню $d_{min} = 3$ та виправляють одиночні помилки. Сюди вхолять коди Хеммінга, Еллаєса, узагальнений код Бергера, лінійний груповий систематичний код.

Код Хеммінга використовується здебільшого в області зберігання даних (фізичні носі пам’яті). Найбільш широко вживається у RAID II, ECC-пам’яті, у яких є вбудовані додаткові біти та відповідні контролери для перевірки на парність контрольні суми.

Лінійнійний груповий систематичний код використовується для коригування переданих повідомлень в системах цифрового зв’язку, телекомунікації, в мережевих протоколах певних рівнів.

Код Еллаєса належить до групи ітеративних кодів. Зважаючи на простоту реалізації володіє високою надмірність порівняно із кодом Хеммінга та лінійнис груповим систематичним кодом.

### 3.3 Код Хеммінга

Для вирішення завдання було вирішено обрати коди Хеммінга, оскільки надмірність такого коду не є значною (на 16 бітів необхідно лише 5 перевірочних бітів). Алгоритм є достатньо простий у реалізації та широковживаний у логічних пристроях зберігання даних.

Якщо оригінальне повідомлення містить $k$ інформаційних розрядів, то у їх комбінаціях формуються $r$ перевірних розрядів. Тому довжина кодової комбінації $n = k + r$. Перевірними розрядами називають лінійні комбінації або зважені суми інформаційних розрядів з вагами 1 і 0.

Кодовим вектором називають бінарну послідовність у кодовій комбінації. Коди Хеммінга володіють також властивостями лінійних кодів. Якщо порахувати суму (різницю) векторів коду, то результуючий вектор також буде належати цьому коду. Коди створюють алгебричну групу поряд з операцією додавання за модулем 2. Мінімальна вага ненульових кодових векторів дорівнює мінімальній кодовій відстані між векторами цього коду.

Мінімальні задані корегувальні властивості кодів визначаються наступним чином: $2^r - 1 = n$.

Основні параметри кодів Хеммінга задаються аргументом $r$. Далі з останнього виразу можна дістати $k$ ($2^r - 1 = k + r$) і порахувати $n$ ($n = k + r$).

Пропорційність величин $k$, $r$ і $n$ для кодів Хеммінга наведено в таблиці 3.1.

Таблиця 3.1 – Співвідношення між кількістю інформаційних, перевірних розрядів і довжиною кодової комбінації для кодів Хеммінга

| $k$ | 1 | 1 | 2 | 3 | 4 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $r$ | 2 | 3 | 3 | 3 | 3 | 4 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 5 |
| $n$ | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

Так як довільне розташування контрольних розрядів ускладнює перевірку прийнятого коду, то для зручності виявлення спотвореного розряду доцільно розміщувати їх на місцях, кратних степені 2. Обрахунок контрольного розряду з номером $i$ буде відбуватись додаванням за модулем 2 (виключна диз'юнкція) наступних $i$ розрядів через кожні $i$ розрядів, починаючи з позиції $i$ та виключаючи контрольні розряди. Положення контрольних розрядів (позначені сірим кольором) та їх визначення ($r1 - r5$) зображено на рисунку 3.1.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | |
| r1 | ⊕ | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | | 0 | | k3 | k5 | k7 | k9 | k11 | k13 | k15 | k17 | k19 | k21 |
| r2 | ⊕ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | | | 0 | | | k3 | k6 | k7 | k10 | k11 | k14 | k15 | k18 | k19 |
| r3 | ⊕ | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | | | 1 | | | k5 | k6 | k7 | k12 | k13 | k14 | k15 | k20 | k21 |
| r4 | ⊕ | 1 | 0 | 0 | 1 | 0 | 1 | 0 | | | | | 1 | | | k9 | k10 | k11 | k12 | k13 | k14 | k15 | | |
| r5 | ⊕ | 1 | 0 | 1 | 0 | 0 | | | | | | | 0 | | | k17 | k18 | k19 | k20 | k21 | | | | |

Рисунок 3.1 – Закодоване повідомлення кодом Хеммінга(21,16)

Виявити чи є помилка в повідомленні (рис. 3.2) можна порівнянням вектору контрольних розрядів, що прийшли у повідомленні, з вектором перерахованих ще раз. Якщо згадані вектори відрізняються, значить можна вважати, що у повідомленні допущена помилка.

| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | | | |
| r1* | ⊕ | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | | 1 | | k3 | k5 | k7 | k9 | k11 | k13 | k15 | k17 | k19 | k21 | |
| r2* | ⊕ | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | | 1 | | | k3 | k6 | k7 | k10 | k11 | k14 | k15 | k18 | k19 | |
| r3* | ⊕ | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | | | 1 | | | k5 | k6 | k7 | k12 | k13 | k14 | k15 | k20 | k21 | |
| r4* | ⊕ | 1 | 0 | 0 | 1 | 0 | 1 | 0 | | | | | 1 | | | k9 | k10 | k11 | k12 | k13 | k14 | k15 | | | |
| r5* | ⊕ | 1 | 0 | 0 | 0 | 0 | | | | | | | 1 | | | k17 | k18 | k19 | k20 | k21 | | | | | |

Рисунок 3.2 – Закодоване повідомлення кодом Хеммінга (21,16) з помилкою

Синдром визначається додаванню по модулю 2 векторів контрольних розрядів. Якщо результуючий вектор вважати числом в двійковій системі числення, то він буде вказувати на номер розряду з помилкою (рис. 3.3).

| | S | r5 | r4 | r3 | r2 | r1 |
|---|---|---|---|---|---|---|
| ⊕ | S п | r5* | r4* | r3* | r2* | r1* |
| | | 0 | 1 | 1 | 0 | 0 |
| | ⊕ | 1 | 1 | 1 | 1 | 1 |
| | | 1 | 0 | 0 | 1 | 1 |
| | | | | | | 19 |

Рисунок 2.3 – Обрахунок синдрому в закодованому повідомленні кодом Хеммінга (21,16)

Якщо в закодованому повідомленні дві помилки, то після процедури виправлення помилок може стати більше ніж було до того, так як дана модель не призначена для одночасного виявлення двох і виправлення одної помилки. Даного недоліку можна позбутись, якщо додати ще один контрольний розряд (розряд подвійного контролю). Тоді довжина кодової комбінації $n = k + r + 1$. Розраховуватись значення цього розряду буде додаванням по модулю 2 всіх розрядів повідомлення і інверсією отриманого результату. Цей розряд означає парність усіх розрядів повідомлення, не включається в загальну нумерацію і ігнорується при підрахунку контрольних розрядів.

Використовуючи дану модель, можна виділити наступні випадки:

1. Синдром нульовий і перерахований розряд подвійного контролю дорівнює вихідному. Якщо помилки в трьох і більше розрядів виключаються, то можна вважати, що помилок немає.

2. Синдром нульовий і перерахований розряд подвійного контролю не дорівнює вихідному. В цьому випадку помилка тільки в розряді подвійного контролю.

3. Синдром ненульовий, перерахований розряд подвійного контролю не дорівнює вихідному. Помилка тільки одна, її виправлення відбувається інверсією біту, на який вказує синдром.

4. Синдром ненульовий, перерахований розряд подвійного контролю дорівнює вихідному. Цей випадок означає подвійну помилку. Виправлення подвійних помилок даний метод не передбачає.

Зазначимо, що з використанням коду Хеммінга дозволить виправляти однократну помилку у вбудованому повідомленні зі заданою довжиною. Не значна надмірність дозволяє використання у зображеннях, не спричиняючи при цьому втрату його якості.

### 3.4 Реалізація комбінованого алгоритму приховування інформації в зображеннях

Реалізований алгоритм кодування Хеммінга (21,16) вбудований в ядро, що описане в розділі 2, та може застосовуватись в комбінації з стеганографічними алгоритмами, що також вбудовані в нього. Було обрано варіант кодування з 16 інформаційними і 5 контрольними бітами.

Кодування Хеммінга складається з операцій кодування та декодування масиву бітів. Максимальна довжина блоку бітів, що підлягає кодуванню можна сконфігурувати константою MaxBlockLength. Також присутні додаткові методи для обрахування кількості контрольних бітів, необхідних для заданої довжини блоку (GetEncodedBitsCount) і для заданої довжини закодованого блоку (GetDecodedBitsCount).

Функція кодування (рис. 3.2) складається з двох основних етапів: побудова масиву бітів з зануленими контрольними бітами та підрахунок значень контрольних бітів парності.

Спочатку відбувається обрахування довжини результуючого масиву та його ініціалізація. Потім для кожного блоку ітеративно обраховується довжина поточного і довжина поточного  закодованого блоку (останній блок може мати меншу довжину). Далі відбувається копіювання бітів блоку з вхідного масив в результуючий так, щоб номери контрольних бітів пропускались. Наступним чином для кожного контрольного біту блоку обраховується його значення. Реалізація цієї функції представлена у лістингу 3.1.

Лістинг 3.1 – Реалізація алгоритму кодування Хеммінга

```
public static BitArray Encode(BitArray bits)
{
    bits = new BitArray(bits);
    var encodedBits = BuildEncodedBitArray(bits);
    return CalculateControlBits(encodedBits, fixErrors: false);
}

private static BitArray BuildEncodedBitArray(BitArray bits)
{
    var encodedBits = new BitArray(GetEncodedBitsCount(bits.Length));

    for (int bitPos = 0, encodedBitPos = 0; bitPos < bits.Length; bitPos += 
MaxBlockLength)
    {
        var blockLength = Math.Min(bits.Length - bitPos, MaxBlockLength);

        if (blockLength <= 0)
        {
            continue;
        }

        var totalBitsCount = GetEncodedBitsCount(blockLength);

        for (int encodedBlockBit = 0, blockBitPos = 0; encodedBlockBit < 
totalBitsCount; encodedBlockBit += 1)
        {
            var encodedBlockBitLog2 = Math.Log(encodedBlockBit + 1, 2);
            if ((int) encodedBlockBitLog2 == (int) 
Math.Ceiling(encodedBlockBitLog2))
            {
                encodedBits.Set(encodedBitPos + encodedBlockBit, false);
            }
            else
            {
                encodedBits.Set(encodedBitPos + encodedBlockBit, bits[bitPos + 
blockBitPos]);
```

```
            blockBitPos += 1;
        }
    }

    encodedBitPos += totalBitsCount;
}

return encodedBits;
  }
```

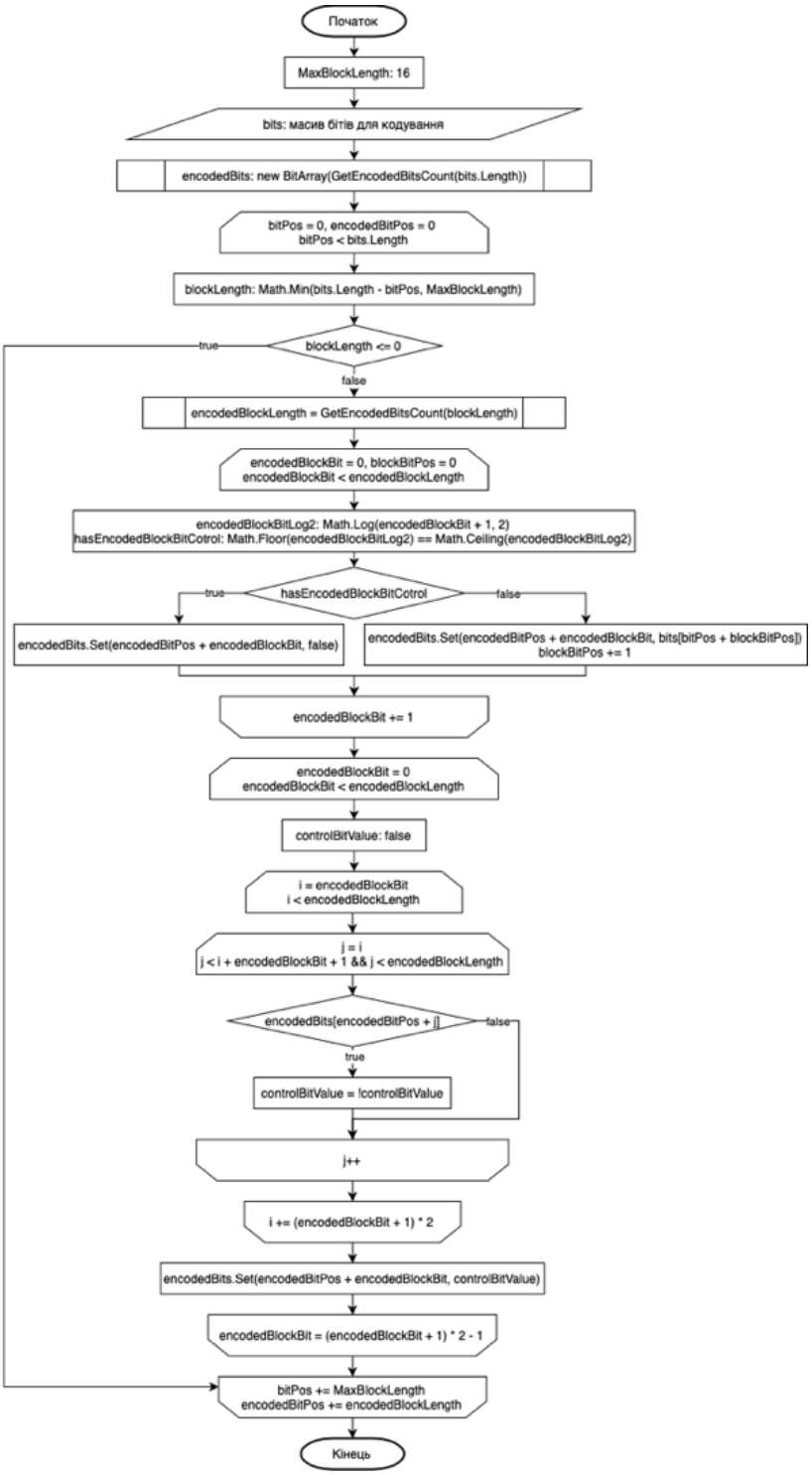



Рисунок 3.4 – Блок схема алгоритму кодування Хеммінга

Суть розрахунків значення контрольного біту зводиться до наступного: біт з номером N буде мати значення «0», якщо кількість бітів з значенням «1», які він контролює, парне, і значення «1», якщо відповідна сума непарна. Під бітами, які контролює біт парності з індексом $N$, маються на увазі всі наступні $N$ біт через кожні $N$ біт, починаючи з позиції $N$.

На рисунку 3.5 червоним кольором виділені біти парності, хрестики ж вказують на біти парності, що контролюють поточний біт.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | |
| X | | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | 1 |
| | X | X | | | X | X | | | X | X | | | X | X | | | X | X | | | 2 |
| | | | X | X | X | X | | | | | X | X | X | X | | | | | X | X | 4 |
| | | | | | | | X | X | X | X | X | X | X | X | | | | | | | 8 |
| | | | | | | | | | | | | | | | X | X | X | X | X | X | 16 |

Рисунок 3.5 – Матриця відповідності бітів закодованого повідомлення та бітів парності

Функція декодування (рис. 3.6) складається трьох етапів: перерахування контрольних бітів, розрахунок синдрому та виправлення помилок, формування результуючого масиву.

Спочатку для кожного блоку зберігаються поточні контрольні біти та обраховуються нові. Якщо контрольний біт змінився, то його індекс додають до поточного значення синдрому. Якщо синдром більший нуля, то він вказує на індекс біту з помилкою. Виправлення такої помилки виконується інвертуванням його значення.

Варто зазначити, що при відсутності помилок синдром буде нульовим, при наявності однієї або двох помилок синдром буде не нульовим та при наявності більше двох помилок блок може виглядати ніби в ньому немає помилок. Також, як відомо, даний алгоритм може вказувати на наявність однієї і двох помилок, але відрізнити ці два випадки не може. Тобто, якщо в блокові дві помилки і синдром не вказує на блок, якого не існує, то невідомо чи буде виправлена хоча б одна з помилок.


Рисунок 3.6 – Блок схема алгоритму декодування Хеммінга

Вкінці формується результуючий масив бітів відкиданням бітів парності (поточний біт є бітом парності, якщо його індекс є степенем двійки). Реалізація описаної функції представлена у лістингу 3.2.

Лістинг 3.2 – Реалізація алгоритму декодування Хеммінга

```
public static BitArray Decode(BitArray encodedBits)
{
    encodedBits = new BitArray(encodedBits);
    encodedBits = CalculateControlBits(encodedBits, fixErrors: true);
    return BuildDecodedBitArray(encodedBits);
}

private static BitArray BuildDecodedBitArray(BitArray encodedBits)
```

```
{
    var maxEncodedBlockLength = GetEncodedBitsCount(MaxBlockLength);

    var decodedBitsCount = GetDecodedBitsCount(encodedBits.Length);
    var decodedBits = new BitArray(decodedBitsCount);

    for (int encodedBitPos = 0, bitPos = 0; encodedBitPos < encodedBits.Length;
encodedBitPos += maxEncodedBlockLength)
    {
        var encodedBlockLength = Math.Min(encodedBits.Length - encodedBitPos,
maxEncodedBlockLength);

        if (encodedBlockLength <= 0)
        {
            continue;
        }

        for (var encodedBlockBit = 0; encodedBlockBit < encodedBlockLength;
encodedBlockBit += 1)
        {
            var encodedBlockBitLog2 = Math.Log(encodedBlockBit + 1, 2);
            if ((int) encodedBlockBitLog2 == (int)
Math.Ceiling(encodedBlockBitLog2))
            {
                continue;
            }

            decodedBits.Set(bitPos, encodedBits[encodedBitPos + encodedBlockBit]);
            bitPos += 1;
        }
    }

    return decodedBits;
}
```

### 3.5 Аналіз результатів роботи реалізованого алгоритму

Результати модуля покращення стійкості стегосистеми, реалізованого у підрозділі 3.4, можна оцінити порівнянням вихідного повідомлення та витягнутого з попередньо модифікованого стегоконтейнера.

На рисунку 3.7 представлений стегоконтейнер, в який вбудовано повідомлення довжиною 200 байт (25% максимальної ємності), що при надлишковості кодування Хеммінга збільшилось приблизно до 265 байт (надлишковість 25%). Після вбудовування повідомлення з ним проводились операції коригування тону, контрасту, конвертації в чорно-білі тони та вибіркові спотворення пензлем в графічному редакторі.

Після порівняння вихідного і витягнутого повідомлення було з’ясовано, що після модифікацій контейнера виникли одиночні помилки в 12 випадкових блоках.

Так як один блок містить 16 інформаційних біт, то виходить, що було спотворено 12 символів повідомлення з повідомлення довжиною 100 символів (12% всього повідомлення). Результатом роботи декодування Хеммінга стало виправлення всіх помилок, так як всі вони були одиночними (втрата 1 біту з блоку довжиною 16 біт).

Результати модифікацій контейнера можна оцінити візуально, порівнявши порожній контейнер (рис. 2.4), стегоконтейнер (рис. 2.8) і модифікований стегоконтейнер в графічному редакторі (рис. 3.7).

Рисунок 3.7 – Заповнений контейнер алгоритмом Коха-Жао і кодуванням Хеммінга (21,16) після модифікацій в графічному редакторі

Таблиця 3.2 – Усереднені кількісні критерії оцінювання стеганосистем з використанням алгоритму Коха-Жао та модифікованого

| Алгоритм | Критерій спотворення зображення | | | | |
|---|---|---|---|---|---|
| | MD | AD | IF | SNR | NC |
| Коха-Жао | 42,5 | 131,057 | 0,994102 | 197,335 | 0,987953 |
| Коха-Жао з кодуванням Хемммінга(21,16) | 22,4 | 67,3 | 0,998712 | 879,45 | 0,99879 |

Таблиця 3.2 містить обраховані критерії спотворення для того ж самого тренувального набору зображень, до якого був застосований алгоритм Коха-Жао та модифікований за допомогою кодування Хеммінга (21,16), використовуючи контейнер, представлений на рисунку 2.4. Аналізуючи отримані дані, можна дійти висновків, що зроблена модифікація не призводить до значних спотворень зображення у порівнянні з вихідним алгоритмом, проте дозволяє відновити та виправити однократні помилки у блоках.

Зауважмо, що розроблена модифікація у комбінації з алгоритмом Коха-Жао показала повне відновлення 12% пошкодженої інформації за рахунок надлишковості в 25% з фактично однаковими критеріями спотворення зображення, що можна вважати задовільним результатом.

### 3.6 Висновки до третього розділу

Основними результатами роботи, представлених у третьому розділі роботи вважаємо наступні:

- детально досліджено завадостійке кодування та обрано код Хеммінга для модифікації існуючого методу приховування інформації з метою запобіганню втрати інформації при спотвореннях спричинених шумом;

- реалізованого модифіковані методи Коха-Жао та псевдовипадкової перестановки з використанням коду Хеммінга (21,6);

- виконано оцінювання основних критеріїв стеганосистеми, яка використовує модифікований алгоритм. Результати порівняння свідчать про доцільність застосування завадостійкого кодування для збільшення ймовірності відновлення вбудованої інформації в разі її пошкодження.

# 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

## 4.1 Охорона праці

Під час розробки програмного забезпечення, що реалізує стеганографічні алгоритми, слід дотримуватись вимог до режиму праці та своєчасного відпочинку під час роботи з ВДТ, ЕОМ та ПЕОМ, що регламентуються Державними санітарними правилами і нормами роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98 (п.5). Щоб запобігти зменшення продуктивності, небажаного травматизму, часткової або повної втрати працездатності працівників варто спланувати графік внутрішньозмінних перерв у роботі для відпочинку.

Дотримання внутрішньозмінного графіку праці та відпочинку забезпечить своєчасне запобігання небажаних та критичних наслідків під час робочого процесу.

Впродовж виконання робіт, у зміні якої більше 50% часу становить оперування ВДТ, ЕОМ та ПЕОМ, потрібно передбачити перерви:

- для вживання їжі (обідня перерва);
- для особистих потреб (згідно з трудовими нормами);
- для потреб, що враховують специфіку трудової діяльності окремих професій.

Обідня перерва регламентується чинним законодавством про працю та Правилами внутрішнього трудового розпорядку підприємства (організації, установи).

Режим праці і відпочинку під час зміни при роботі з ВДТ, ЕОМ та ПЕОМ розробляється згідно з типом трудової діяльності, важкості та напруженості праці окремо для кожного виду діяльності.

Розрізняють три професійні групи за характером трудової діяльності згідно з діючим класифікатором професій (ДК - 003 - 95 і Зміна №1 до ДК - 003 - 95) класу професіоналів в галузі обчислень та комп'ютеризації:

1. Професіонали в галузі обчислювальних систем виконують роботу, що пов'язана з опрацюванням інформації, яка надходить від ВТД по запиту або періодично, та супроводжується періодичними переривами змінної тривалості. Робота характеризується високим навантаженням на зір, емоційним напруженням та виконується у довільному темпі.

2. Професіонали в галузі програмування працюють переважно з ЕОМ та документацією, що характеризується статичним положенням тіла під час роботи, періодичним високим навантаженням на кисті рук, концентрацією уваги на статичну точку та інтенсивною розумовою діяльністю. Робота відбувається у оперуванні ЕОМ з прийняттям рішень в умовах браку часу.

3.Робота професіоналів в інших галузях обчислень та комп'ютеризації полягає у одноманітних за характером маніпуляцій за ЕОМ, швидкому введенню даних з клавіатури та характеризується високим навантаженням на кисті рук, напруженням зору та емоційним навантаженням.

Під час роботи з ЕОМ залежно від характеру праці встановлюють наступні внутрішньозмінні режими праці та відпочинку при 8-годинній денній робочій зміні:

- для операторів в галузі обчислювальних систем потрібно встановити регламентовані перерви на 15 хвилин з періодом в дві години;

- для операторів в галузі програмування потрібно встановити регламентовані перерви на 15 хвилин з періодом в одну годину, враховуючи тільки час роботи за ВДТ;

- для операторів в інших галузях обчислень та комп'ютеризації потрібно встановити регламентовані перерви на 10 хвилин з періодом в одну годину, враховуючи тільки час роботи за ВДТ.

Безперервна робота з ВДТ повинна продовжуватись не більше 4-х годин для всіх видів трудової діяльності, якщо неможливо застосовувати регламентовані перерви.

У випадку 12-годинної робочої зміни регламентовані перерви перші 8 годин слід проводити згідно з режимами праці та відпочинку при 8-годинній денній

робочій зміні, а до кінця зміни для всіх видів трудової діяльності перерви повинні становити 15 хвилин з періодом в одну годину.

Під час роботи з ВДТ, ЕОМ та ПЕОМ варто розподілити перерви таким чином, щоб у деяких з них була можливість приділити увагу зміні типу діяльності на фізичну, а саме виконання комплексу гімнастичних вправ, спрямованих на розслаблення м'язів, зняття втоми очей та поліпшення мозкової працездатності.

У випадку високого рівня напруженості роботи з ВДТ обладнують спеціальні кімнати для психологічного відпочинку, відвідувати які варто під час регламентованих перерв або в кінці робочого дня.

Для приміщень з ЕОМ, де відбувається розробка програмного забезпечення, що реалізує стеганографічні алгоритми, слід обрати фарбу світлого холодного кольору для стін, що візуально збільшить розміри приміщення та загальну його освітленість. Поверхня підлоги повинна бути рівною та такою, що виключатиме небезпеку ковзання. Освітлення приміщень повинно відповідати вимогам ДБН В.2.5-28:2018 «Природне і штучне освітлення». В першу чергу необхідно створити такі умови, щоб освітленість на робочій поверхні відповідала характеру зорової роботи і знаходилась в межах встановлених норм.

При розробці програмного забезпечення, що реалізує стеганографічні алгоритми, користуються лініями електромереж. Для побудови системи потрібно використовувати лише якісні та сертифіковані пристрої та засоби. ВДТ, ЕОМ, ПЕОМ та периферійні пристрої повинні бути підключені до електромережі виключно за допомогою справних штепсельних з’єднань і електророзеток, в яких повинні бути передбачені контакти для підключення нульового захисного провідника. Важливим є те, що спочатку повинно відбуватись приєднання нульового захисного провідника, а потім фазового та нульового робочого провідників. Усі електроприлади, згідно з ДНАОП 0.00-1.21-98, повинні бути заземленні за допомогою нульового захисного провідника.

Передбачені заходи по забезпеченню безпеки, виробничої санітарії, гігієни праці та пожежної безпеки для приміщення, в якому проводиться порівняльний

аналіз стеганографічних алгоритмів приховування інформації в зображеннях, забезпечують безпечні та комфортні умови праці.

### 4.2 Безпека об’єктів виробничого призначення у воєнний час

На основі вивчення факторів, що впливають на довговічність об'єктів, та оцінки стійкості елементів та виробництв до шкідливих факторів ядерної, хімічної та біологічної зброї, стихійних лих та промислових аварій, існує необхідність проведення організаційних, технічних і технологічних заходів щодо підвищення стабільності роботи.

Вживання технологічних заходів підвищує стабільність установ шляхом зміни технологічних процесів та режимів, можливих в надзвичайних ситуаціях. В тому числі існують загальні організаційні інженерно-технічні заходи, які мають проводитись на всіх об'єктах.

1. Забезпечення захисту людей та їх життєдіяльності. Створення на об'єкті надійної системи оповіщення про загрозу нападу противника, радіоактивне забруднення, хімічне і біологічне зараження, загрозу стихійного лиха і виробничої аварії. Організація розвідки і спостереження за радіоактивним забрудненням, хімічним і біологічним зараженням; гідрометеорологічне спостереження за рівнем води, напрямком і швидкістю вітру, рухом і поширенням хмари радіоактивного забруднення.

Підготовка до евакуації населення, розміщеного в зонах можливих руйнувань і катастрофічного затоплення. Завчасна підготовка місць евакуації, організація прийому евакуйованого населення на територію населених пунктів.

Постачання населення продуктами харчування, питною водою, предметами першої необхідності; комунальне побутове обслуговування населення з урахуванням проведення евакуаційних заходів, забезпечення захисту продовольчих запасів.

Завчасна підготовка до масової санітарної обробки населення і знезаражування одягу, організація взаємодії з установами охорони здоров'я для медичного обслуговування населення у надзвичайних ситуаціях.

Навчання населення способам захисту, надання першої допомоги, практичним діям в умовах надзвичайних ситуацій, морально-психологічна підготовка населення для виживання.

Забезпечення чіткої інформації про обстановку та правила дій і поведінки населення в надзвичайних ситуаціях мирного і воєнного часу.

2. Захист цінного й унікального устаткування. Захистити цінне і унікальне устаткування можна завдяки проведенню інженерно-технічних заходів, щоб зменшити небезпеку пошкодження і руйнування цінного й унікального устаткування, станків з програмним керуванням, шліфувальних, токарних, розточувальних, зубофрезерних, пресових станків, автоматичних конвеєрних ліній та іншого устаткування.

Варіантами такого захисту є розміщення зазначеного устаткування в заглиблених приміщеннях а також використання спеціальних захисних пристосувань, закріплення станків на фундаментах, застосування контрфорсів для підвищення стійкості проти перекидання обладнання.

3. Стійкість роботи галузі рослинництва. Планування і проведення заходів захисту сільськогосподарських рослин, урожаю в різних надзвичайних ситуаціях.

Встановлення надійної взаємодії зі станцією захисту рослин, радіологічною і агрохімічною лабораторією для організації спостереження за зараженістю посівів сільськогосподарських культур та ґрунтів, відбір необхідних проб та їх аналіз.

Впровадження у виробництво високоврожайних, стійких проти небезпечних хвороб і шкідників сільськогосподарських культур.

Підготовка техніки і хімічних засобів захисту сільськогосподарських культур від біологічних засобів ураження.

Розробка заходів збирання урожаю в умовах обмеженості забезпечення людьми, технікою, паливом і мастилами, порушення міжгалузевих зв'язків, технології доведення урожаю до кондиції.

Організація зберігання і переробки урожаю в господарстві при порушенні зв'язків із заготівельними й переробними організаціями та підприємствами.

Розробка і підготовка до впровадження спрощених технологій вирощування сільськогосподарських культур, підготовка до зміни сівозмін і перепрофілювання рослинництва.

Забезпечення ефективного використання сільськогосподарських угідь в умовах радіоактивного забруднення, зараження хімічними і біологічними засобами.

Підготовка всіх засобів для захисту працюючих у рослинництві в різних умовах надзвичайних ситуацій.

4. Стійкість роботи тваринництва. Підготовка до проведення ветеринарно-санітарних заходів, спрямованих на зниження втрат тварин від сучасних засобів ураження. Завчасна підготовка приміщень для утримання тварин. Розробка заходів захисту тварин на пасовищах. Створення запасів кормів і організація забезпечення водою.

Організація ветеринарної розвідки в господарстві, відбір необхідних проб та їх аналіз.

Створення індивідуальних засобів захисту для елітного поголів'я худоби.

Розробка заходів евакуації тварин із зон можливих руйнувань, катастрофічного затоплення, районів хімічного зараження, підготовка місць для евакуації тварин. Планування заходів захисту кормів, джерел водопостачання і тваринницьких ферм.

Організація забезпечення основних виробничих процесів у тваринництві електроенергією від автономних джерел електропостачання, у разі відключення від центральної енергомережі.

Підготовка до постійної готовності спеціальної техніки для обробки тварин, а також пристосування для цієї мети іншої техніки, наявної в господарстві.

Організація ветеринарної обробки, утилізації і забою уражених тварин, тимчасового зберігання м'ясної продукції при порушенні господарських зв'язків із заготівельними організаціями і підприємствами.

Розробка найпростіших технологій переробки і зберігання продукції тваринництва в разі неможливості відправки переробним підприємствам і реалізації.

Організація забезпечення працюючих у тваринництві колективними та індивідуальними засобами захисту.

5. Підвищення стійкості мереж комунального господарства. Для забезпечення стійкості роботи об'єктів повинні проводитись інженерно-технічні заходи на мережах комунального господарства з метою захисту джерел тепла із заглибленням у ґрунт комунікацій. Котельні слід розміщувати в спеціальному окремо розміщеному приміщенні.

Якщо об'єкт одержує тепло з міської теплоцентралі, необхідно провести заходи для забезпечення стійкості трубопроводів і розподільних пристроїв, підведених до об'єкта.

Теплова мережа має будуватися за кільцевою системою з прокладанням труб у спеціальних каналах зі з'єднанням паралельних ділянок. Ці пристосування необхідно розміщувати в оглядових колодязях, на території, що не завалюється при руйнуванні будівель.

Система каналізації має будуватись окремо: одна для дощових, друга для промислових і господарських вод. На об'єкті має бути не менше двох виводів з підключенням до міських каналізаційних колекторів, а також виводи і колодязі з аварійними засувками на об'єктових колекторах з інтервалом 50 м на території, що не завалюється, для аварійного скидання неочищеної води в найближчі штучні та природні заглиблення.

На деяких промислових об'єктах є системи для забезпечення технології виробництва: для подання кисню, аміаку, стиснутого повітря та інших рідких і газових реактивів. Для цих систем розробляють заходи для попередження виникнення вторинних факторів зброї, стихійних лих та виробничих аварій і катастроф.

6. Забезпечення стійкості роботи паливно-енергетичного комплексу і водопостачання. Створення резерву енергетичних потужностей за рахунок

автономних пересувних електростанцій, а також місцевих джерел електроенергії. Підготовка автономних електростанцій до роботи за спеціальним режимом (графіком) для забезпечення технологічних процесів виробництва, для яких неможливі тривалі перерви в електропостачанні.

З метою попередження аварій на електричних мережах необхідно установити автоматичну систему відключення при виникненні перенапруги. Повітряні лінії електропостачання замінити на підземно-кабельні.

Створення необхідних запасів (резервів) паливно-мастильних матеріалів та інших видів палива й організація їх безпечного зберігання.

Щоб не допустити зупинки підприємства через дефіцит палива, необхідно підготуватись для роботи на різних видах палива: нафта, вугілля, газ. Для підвищення стійкості забезпечення водою слід провести такі заходи. Необхідно створити основні і резервні джерела водопостачання. Як резервне джерело краще мати артезіанську свердловину, яку необхідно підключити до системи водопостачання.

# ВИСНОВКИ

У роботі детально розглянуто основні етапи становлення стеганографії як науки, напрямки сучасної стеганографії, проаналізовано основні стеганографічні методи приховування інформації в графічних зображеннях.

В результаті виконаного порівняльного аналізу на основі обраних кількісних характеристик: AD, MD, IF, SNR, NC, прийняття рішення з використанням методу аналізу ієрархій щодо оптимального методу приховування інформації в графічних зображеннях було з'ясовано основні переваги та виділено ряд недоліків для кожного з них. На основі виявлених даних було вирішено скомбінувати приховування інформації у контейнер-зображення з завадостійким кодом Хеммінга для запобігання втрат частини даних при різних модифікаціях стегоконтейнера.

Оскільки особливості стегосистем не захищають в повній мірі від втрати інформації після різноманітних модифікацій в графічних редакторах, то можливість відновити прочитані дані за рахунок їх надлишковості є універсальним способом, який підходить для багатьох рішень. Для експерименту було проведено різноманітні модифікації стегоконтейнера в графічному редакторі після вбудовування даних методом Коха-Жао. Результатом стало те, що втрачені 12% інформації були відновлені завдяки кодуванням Хеммінга при надлишковості в 25%.

Зважаючи на складність спроектованого програмного забезпечення було вирішено використовувати архітектурні шаблони, що дозволили спростити процес розширення функціоналу і реалізувати продуктивне рішення мінімальною кількістю рядків коду. В рамках опису роботи розробленої архітектури побудованої системи наведено детальний опис розробленого алгоритму у вигляді блок схем.

Враховані основні вимоги до режиму праці та своєчасного відпочинку під час розробки програмного забезпечення.

Зважаючи на вище все наведене можна дійти висновку, що в результаті проведеної роботи вдалось дійти поставленої мети в повному обсязі.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Шелест М.Є., Андреев В.1. Комп'ютерна стеганографія та ЇЇ можливості / М.Є. Шелест, В.1. Андреев // Сучасна спеціальна техніка № 1 (24), 2011. С.97-104 [Електронний ресурс]. Режим доступу: https://bit.ly/2VDsink.

2. О. В. Генне, ТОВ "Конфідент" журнал "Захист інформації. Конфідент", № 3, 2000.

3. Кузнецов О. О. К89 Стеганографія: навчальний посібник / О. О. Кузнецов, С. П. Євсеєв, О. Г. Король. - Х.: Вид. ХНЕУ, 2011. [Електронний ресурс] – Режим доступу до ресурсу: https://bit.ly/3mq3s6c.

4. Конахович Г.Ф. Комп’ютерна стеганографія. Теорія і практика / Г.Ф. Конахович, А. Ю. Пузиренко. - Київ: МК-Пресс, 2006. – 288с.

5. Востриков А.С., Пустовой Н.В. "Цифровая обработка изображений в информационных системах"; Учебник НГТУ, Новосибирск 2002.

6. Рябко Б. Я., Фионов А. Н. Основы современной криптографии и стеганографии. 2-е изд. / Рябко Б. Я., Фионов А. Н. // М.: Горячая линия - Телеком, 2013. 232 с.

7. В.Г. Бабенко, В.М. Зажома, О.Б. Нестеренко. Метод вбудовування стегоповідомлення на основі ключового елемента / В.Г. Бабенко, В.М. Зажома, О.Б. Нестеренко. // Захист інформації. 2014. С. 53-58.

8. Sanmitra I., Shivananda P., Shrikant B., Usha B, Image Steganography using Sudoku Puzzle for Secured Data Transmission // International Journal of Computer Applications (0975 – 888) Volume 48– No.17, June 2012.

9. Бернет С., Пейн С.: Криптография. Официальное руководство RSA Security – М. «Бином», 2012. – 325 с.

10. Аграновский А.В. Основы компьютерной стеганографии / А.В. Аграновский, П.Н. Девянин, Р.А. Хади, А.В. Черемушкин. –М: Радио и связь, 2003. –152 с.27.

11. Wang, H. & Wang, S. 2004. Cyber warfare: Steganography vs. steganalysis. Communications of the ACM, 47(10):76-82.

12. Mazdak Zamani and Azizah Abdul Manaf. “Azizah’s Formula to Measure the Efficiency of Steganography Techniques”. 2nd International Conference on Information and Multimedia Technology (ICIMT 2010). December 28-30, 2010. Hong Kong, China.

13. Mazdak Zamani, Hamed Taherdoost, Azizah Abdul Manaf, Rabiah Ahmad, and Akram Zeki. “Robust Audio Steganography via Genetic Algorithm”. Third International Conference on Information & Communication Technologies ICICT2009. ISBN: 9781424446087. Pages 149 - 153. 15-16 August 2009. Karachi, Pakistan.

14. 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. 351–355.

15. Low Cost Spatial Watermarking / Darmstaedter V., Delaigle J., Quisquater J., Macq B. // Computers and Graphics. – USA, New York. – 1998. – Vol. 22, Issue 4. – Pp. 417- 424.

16. Сачик Т.В. Захист персональної інформації в задачах аналізу та обробки великих даних : дипломна робота магістра за спеціальністю „125 — кібербезпека“/ Т.В. Сачик. — Тернопіль: ТНТУ, 2019. — 107 с.

17. Гулка Ю.І. Порівняльний аналіз стеганографічних методів з контейнером-зображенням : Дипломна робота магістра за спеціальністю „125 — кібербезпека“/ Ю.І. Гулка. — Тернопіль: ТНТУ, 2019. — 123 с.

18. Кінах Я. І. Застосування методів оптичних обчислень та програмного забезпечення для задач криптографічного захисту інформації / Я. І. Кінах // Матеріали І науково-технічної конференції факультету комп’ютерно-інформаційних систем і програмної інженерії „Інформаційні моделі, системи та технології“, секція: програмна інженерія та моделювання складних розподілених систем, 20 травня 2010 р. : тези доповідей – Тернопіль : ТНТУ, 2010. – С. 58.

19. Чертова М. Захист інформації методами комп’ютерної стеганографії / Чертова М. // Збірник тез VIII всеукраїнської студентської науково-технічної конференції „Природничі та гуманітарні науки. Актуальні питання“, 23-24 квітня 2015 р. — Т. : ТНТУ, 2015 — Том 1. — С. 104. — (Секція: Інформаційні технології).

20. Грибан В. Г., Негодченко О. В. Охорона праці: навчальний посібник. -2-е видання. Київ: Центр учбової літератури, 2018.- 280 с, ISBN 978-966-364-832-3.

21. Запорожець О. І., Протоєрейський О. С., Франчук Г. М., Боровик І. М. Основи охорони праці підручник Київ: Центр учбової літератури, 2017. - с.264, ISBN 978-617-673-423-9.

22. М. С. Одарченко, А. М. Одарченко, В. І. Степанов, Я. М. Черненко. Основи охорони праці: підручник/ – Х. : Стиль-Издат, 2017. – 334 с. ISBN 966-7885-84-4.

23. Охрана окружающей среды: учеб. для техн. спец. вузов./ С.В. Белов, Ф.А. Барбинов, А.Ф. Козьяков и др. ; под ред. С.В.Белова М.: Высшая школа, 1991.- 319с. ISBN 5-06-000665-1.

24. Васійчук В.О., Гончарук В.Є., Качан С.І., Мохняк С.М.Основи цивільного захисту: Навч. посібник / В.О. Васійчук, В.Є Гончарук, С.І.Качан, С.М. Мохняк.- Львів: Видавництво Національного університету "Львівська політехніка", 2010.- 417с.

25. Білявський Г. О. Основи екології: підручник для студ. вищих навч. закладів / Г. О. Білявський, Р. С. Фурдуй, І. Ю. Костіков. К. : Либідь, 2004. - 408 с. ISBN 966-06-0289-8.

26. Запольський А.К. Основи екології: підр. для студ. техн. технол. спец. вищ. навч. закл. / А. К. Запольський, А.І. Салюк; за ред. К.М. Ситника. К.: Вища школа, 2001.- 358с. ISBN 966-642-059-7.

27. Тарасова В.В. Екологічна статистика // Київ: «Центр учбової літератури», 2008 р.-391с.

# ДОДАТКИ

ДОДАТОК А - ТЕХНІЧНЕ ЗАВДАННЯ

Міністерство освіти і науки України

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра програмної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедрою

програмної інженерії

"___ "_____ 2020 р.

# ТЕХНІЧНЕ ЗАВДАННЯ

на виконання кваліфікаційної роботи

на тему: «Порівняльний аналіз стеганографічних алгоритмів приховування інформації в зображеннях»

Керівник роботи:

д.ф.-м.н., професор Петрик М. Р.

"___" _____ 2020 р.

_____

Виконавець:

студент групи СПм-61

Резнік Дмитро Володимирович

"___" _____ 2020 р.

_____

Тернопіль 2020

**ЗМІСТ**

# 1 ПІДСТАВИ ДО РОЗРОБКИ

Розробка проводиться у відповідності до графіку навчального плану на 2020 рік, та згідно наказу на виконання кваліфікаційної роботи студента.

Тема роботи: «Порівняльний аналіз стеганографічних алгоритмів приховування інформації в зображеннях».

# 2 ПРИЗНАЧЕННЯ РОЗРОБКИ

Програмне забезпечення являє собою ядром з інтегрованими реалізаціями стеганографічних алгоритмів та алгоритмів завадостійкого кодування.

Метою даної роботи є розробка комбінованого алгоритму приховування інформації в зображеннях на основі існуючих після їх порівняльного аналізу.

Одним з ключових моментів є забезпечення легкості інтеграції системи в інші програмні продукти завдяки простоті інтерфейсу взаємодії.

За результатами виконаної роботи необхідно розробити програмне забезпечення, яке вирішить задачу прихованої передачі інформації без безпосереднього використання стегосистеми з покращенням її стійкості до втрати даних.

# 3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

## 3.1 Функціональні характеристики

Програмне забезпечення має виконувати наступні дії:

- вбудовувати в зображення повідомлення;
- діставати повідомлення з зображення;
- попередньо кодувати вбудовуване повідомлення;
- підтримувати можливість обрати алгоритм вбудовування;
- перевіряти можливість вбудовування повідомлення в зображення.

## 3.2 Параметри технічних засобів

ПК з 4 ГБ оперативної пам’яті, встановленою системою Windows 7 і вище, MacOS 10.13 вище, не менше 5 ГБ вільного місця на жорсткому диску. Процесор з тактовою частотою від 1.4 GHz і більше.

Наявність встановленої бібліотеки .NET Core 2.2.

## 3.3 Інформаційна та програмна сполучність

Програмний продукт повинен коректно функціонувати в операційних системах Windows 7 та новіших, на яких доступна для встановлення бібліотека .NET Core 2.2. Розроблювана бібліотека класів повинна бути пристосована до використання у інформаційних системах та програмних засобах. Розробку виконувати з використанням бібліотек та технологій мови C# в середовищі програмування JetBrains Rider 2020.2 з використанням технології .NET.

## 4 СТАДІЇ РОЗРОБКИ

В ходів реалізації робота повинна пройти крізь наступні стадії розробки:
аналіз предметної області;

- проектування архітектури;
- реалізація архітектури;
- реалізація інтерфейсу;
- тестування результатів розробки;
- оформлення супровідної документації;
- здача роботи.

## 5 ПРОГРАМНА ДОКУМЕНТАЦІЯ

Для програмного продукту повинні бути розроблені наступні документи:

- Пояснювальна записка;
- Технічне завдання;
- Презентаційний матеріал;
- Додатки.

## 6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Розроблений програмний продукт має виконувати всі вимоги, що складаються з перерахованих у п. 3.1 характеристик.

Приймання проводиться спеціально створеною екзаменаційною комісією в термін до:

«__» грудня 2020р.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**
**ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**
**ІМЕНІ ІВАНА ПУЛЮЯ**

**М А Т Е Р І А Л И**

**VIII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ**

# «ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

**9–10 грудня 2020 року**

**ТЕРНОПІЛЬ**
**2020**

УДК 001
М34

## ПРОГРАМНИЙ КОМІТЕТ

**Голова:** Лупенко Сергій Анатолійович – докт. техн. наук, професор.

**Співголови:** Марущак Павло Орестовищ – проректор з наукової роботи, докт. техн. наук, професор.
Баран Ігор Олегович – канд. техн. наук, доцент, декан факультету ФІС.

**Науковий секретар:** Семенишин Галина Мирославівна – старший викладач.

**Члени:** докт. фіз.-мат. наук, професор В. Кривень; докт. техн. наук, професор М. Приймак; канд. техн. наук, доцент, Г. Осухівська; докт. техн. наук, професор М. Карпінський; канд. пед. наук, доцент Ж. Баб'як; докт. фіз.-мат. наук, професор М. Петрик; канд. техн. наук, доцент Н. Загородна.

## ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

**Голова:** Скоренький Юрій Любомирович – канд. техн. наук, доцент.

**Члени:** канд. екон. наук, доцент І. Струтинська; канд. техн. наук, доцент Я. Кінах; асистент М. Стадник; асистент Н. Шаблій; ст. викладач Л. Джиджора.

М34 Матеріали VIII науково-технічної конфції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 9 – 19грудня 2020р.). – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2020. – 196 с.

**Адреса оргкомітету:** ТНТУ ім. І. Пулюя, м. Тернопіль, вул. Руська, 56, 46001,
тел. (0352) 52-41-33, факс (0352) 254983.
E-mail: conffis2020@gmail.com
Редагування, оформлення, верстка: Семенишин Г.М.

## СЕКЦІЇ КОНФЕРЕНЦІЇ, ЯКІ ПРЕДСТВЛЕНІ В ЗБІРНИКУ

- Математичне моделювання;
- Інформаційні системи та технології;
- Комп'ютерні системи та мережі;
- Програмна інженерія та моделювання складних розподілених систем;
- Новітні фізико-технічні та освітні технології.

В збірнику надруковано тези доповідей VIII науково-технічної конференції «Інформаційні моделі, системи та технології» (Тернопіль, 9–10 грудня 2020 р.) за такими науковими напрямками: математичне моделювання; інформаційні системи та технології; комп'ютерні системи та мережі; програмна інженерія та моделювання складних розподілених систем; новітні фізико-технічні та освітні технології.

Розрахований на науковців, викладачів та студентів вузів.

**За зміст тез та дотримання норм академічної доброчесності відповідальність несе автор**.

УДК 004.056
**Д.В. Резнік – магістрант**
(Тернопільський національний технічний університет ім. І. Пулюя, Україна)

# ПОРІВНЯЛЬНИЙ АНАЛІЗ СТЕГАНОГРАФІЧНИХ АЛГОРИТМІВ ПРИХОВУВАННЯ ІНФОРМАЦІЇ В ЗОБРАЖЕННЯХ

UDC 004.056
**D.W. Reznik – graduate student**

# COMPARATIVE ANALYSIS OF STEGANOGRAPHIC ALGORITHMS FOR HIDE INFORMATION IN IMAGES

Стеганографія – наука про методи захисту інформації шляхом приховування факту її існування в певному середовищі. Приховування факту існування таємного повідомлення завжди видавалося доцільним для його захисту, а наявність різних технічних, хімічних, фізичних і психологічних методів такого приховування забезпечувало можливість його реалізації.

Стеганографічна система або стегосистема – це сукупність засобів та методів, що використовуються для забезпечення прихованого каналу передачі даних[3]. Основними елементи узагальненої стеганографічної системи зображено на рис. 1.


Рисунок 1. Елементи узагальненої стеганографічної системи

Контейнер – будь-яка інформація, призначена для приховування таємних повідомлень[1]. Порожній контейнер – контейнер без вбудованого повідомлення. Заповнений контейнер або стегоконтейнер – контейнер, що містить вбудовану інформацію[2]. Вбудоване (приховане або секретне) повідомлення – повідомлення, яке вбудоване в контейнер. Стегоключ або просто ключ – секретний ключ, необхідний для приховування інформації[3].

При приховуванні даних у нерухомих зображеннях можливі такі методи:

- приховування даних у просторовій області;
- приховування даних в частотній області;
- розширення спектру.

Приховування даних у просторовій області може здійснюватися за допомогою наступних методів:

- метод заміни найменш значущого біта;
- метод псевдовипадкового інтервалу;
- метод псевдовипадкової перестановки;
- метод блокового приховування;
- метод заміни палітри;
- метод квантування зображення;

- метод Куттера-Джордана-Боссена;
- метод Дармстедтера-Делейгла-Квісквотера-Макка.

Приховування даних в частотній області можливе при використанні таких методів:

- метод відносної заміни величин коефіцієнтів дискретно косинусногоперетворення;
- метод Бенгама-Мемона-Ео-Юнга;
- метод Хсу і Ву;
- метод Фрідріха.

Оскільки, існує значна кількість стеганографічних методів приховування інформації у зображеннях, тому необхідно дослідити переваги та недоліки кожного з них, з метою ідентифікації доцільності застосування того чи іншого методу в конкретній ситуації та умовах.

Для виконання порівняльного аналізу стеганографічних методів приховування інформації у зображеннях було обрано наступні критерії:

- стійкість системи до модифікації контейнера;
- ефективність для стеганоаналізу;
- максимальний об’єм приховуваних даних.

Стеганографічні методи приховування інформації в зображеннях є доцільними при забезпеченні обмеженого доступу до інформації та її захисту. Розуміння того чи іншого методу, його переваги аз певних умов застосування сприятиме досягненню надійного рівня безпеки та захисту.

**Література**

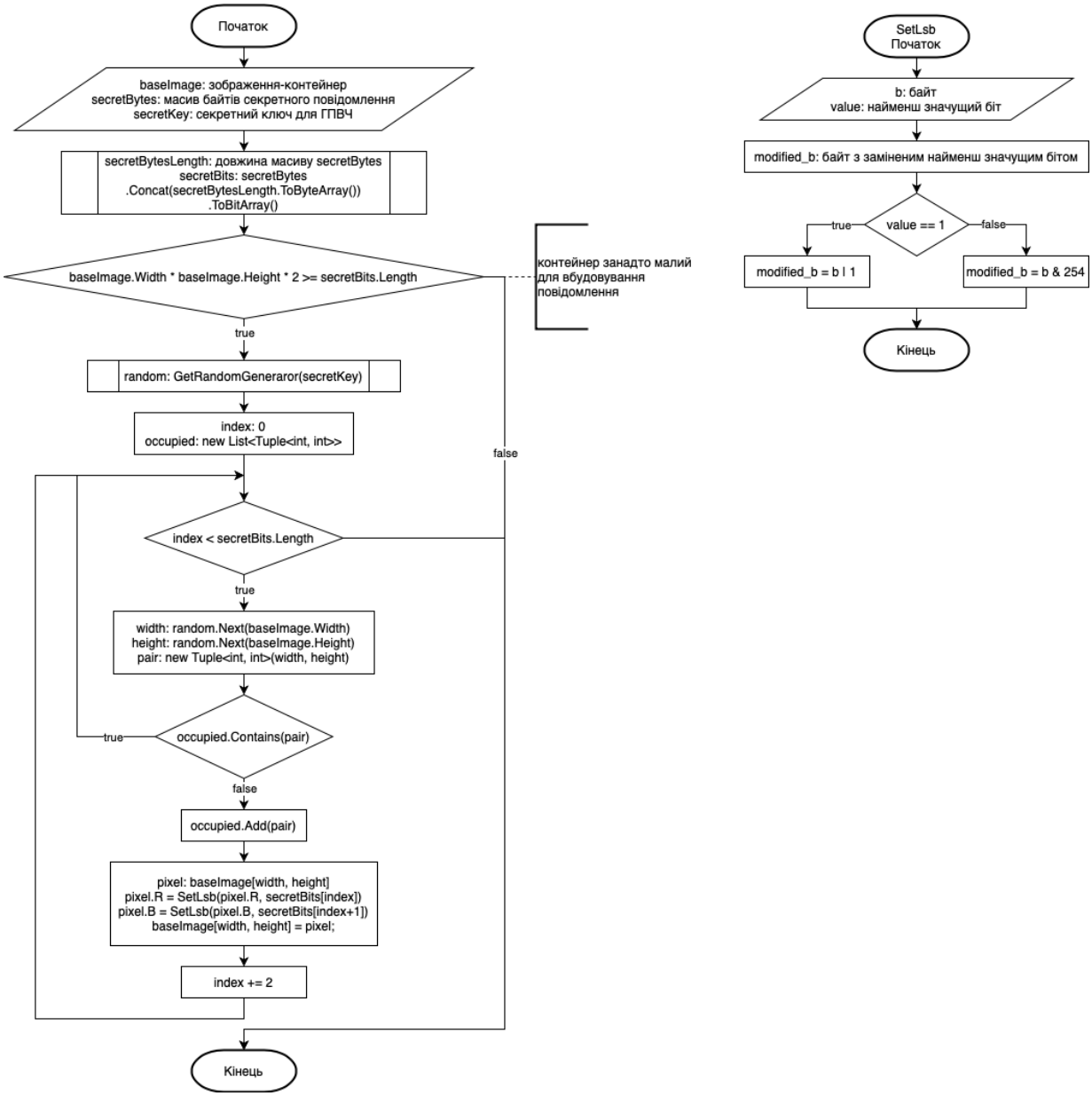
1. Кузнецов О. О. К89 Стеганографія: навчальний посібник / О. О. Кузнецов, С. П. Євсеєв, О. Г. Король. - Х.: Вид. ХНЕУ, 2011. - 232с. [Електронний ресурс] – Режим доступу до ресурсу: https://bit.ly/3mq3s6c.
2. Конахович Г.Ф. Комп’ютерна стеганографія. Теорія і практика / Г.Ф. Конахович, А. Ю. Пузиренко. - Київ: МК-Пресс, 2006. – 288с.
3. О. В. Генне, ТОВ "Конфідент" журнал "Захист інформації. Конфідент", № 3, 2000.

# ДОДАТОК В

## Електронні документи

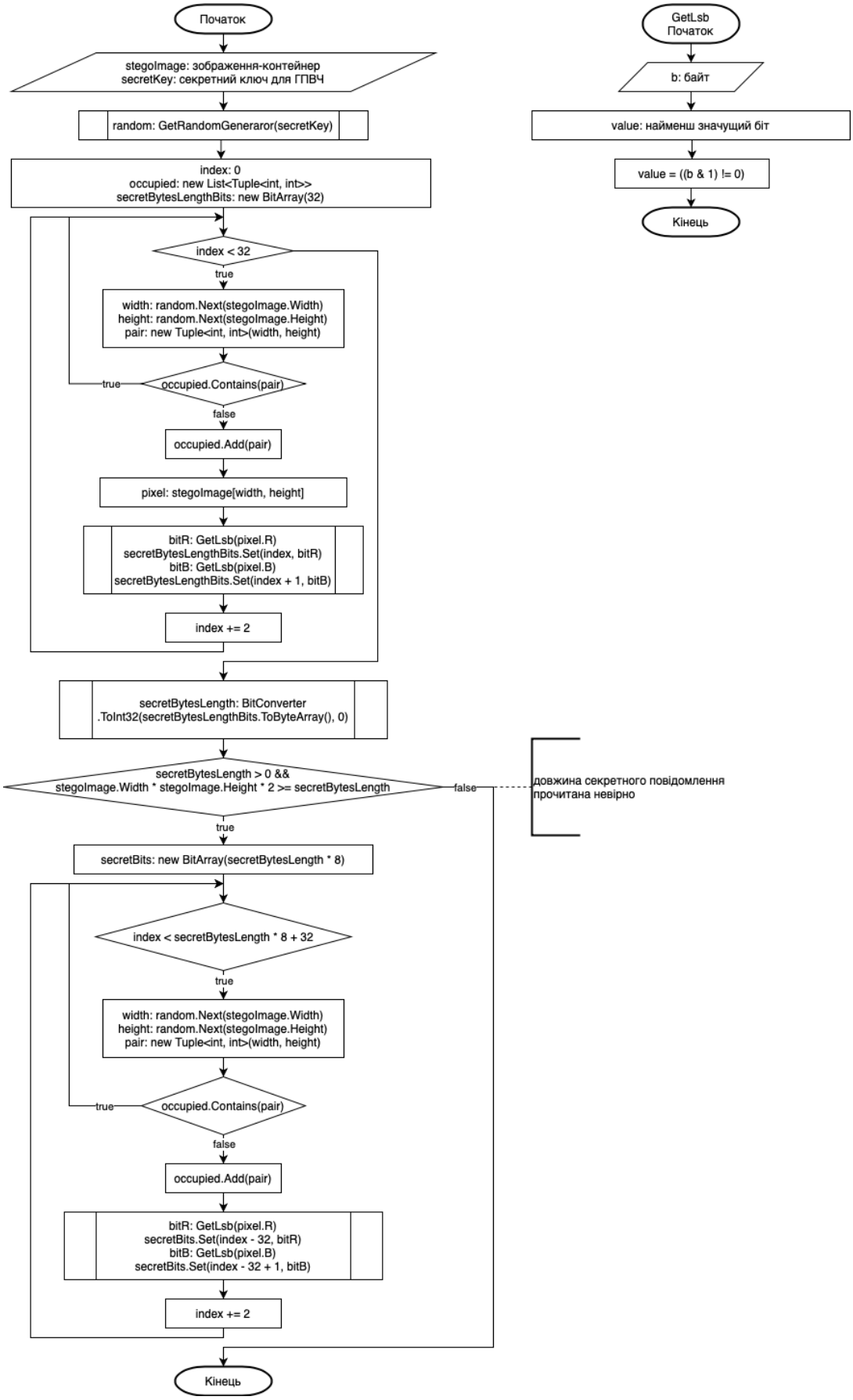
# ДОДАТОК Г

## Блок схема алгоритму вбудовування методом НЗБ



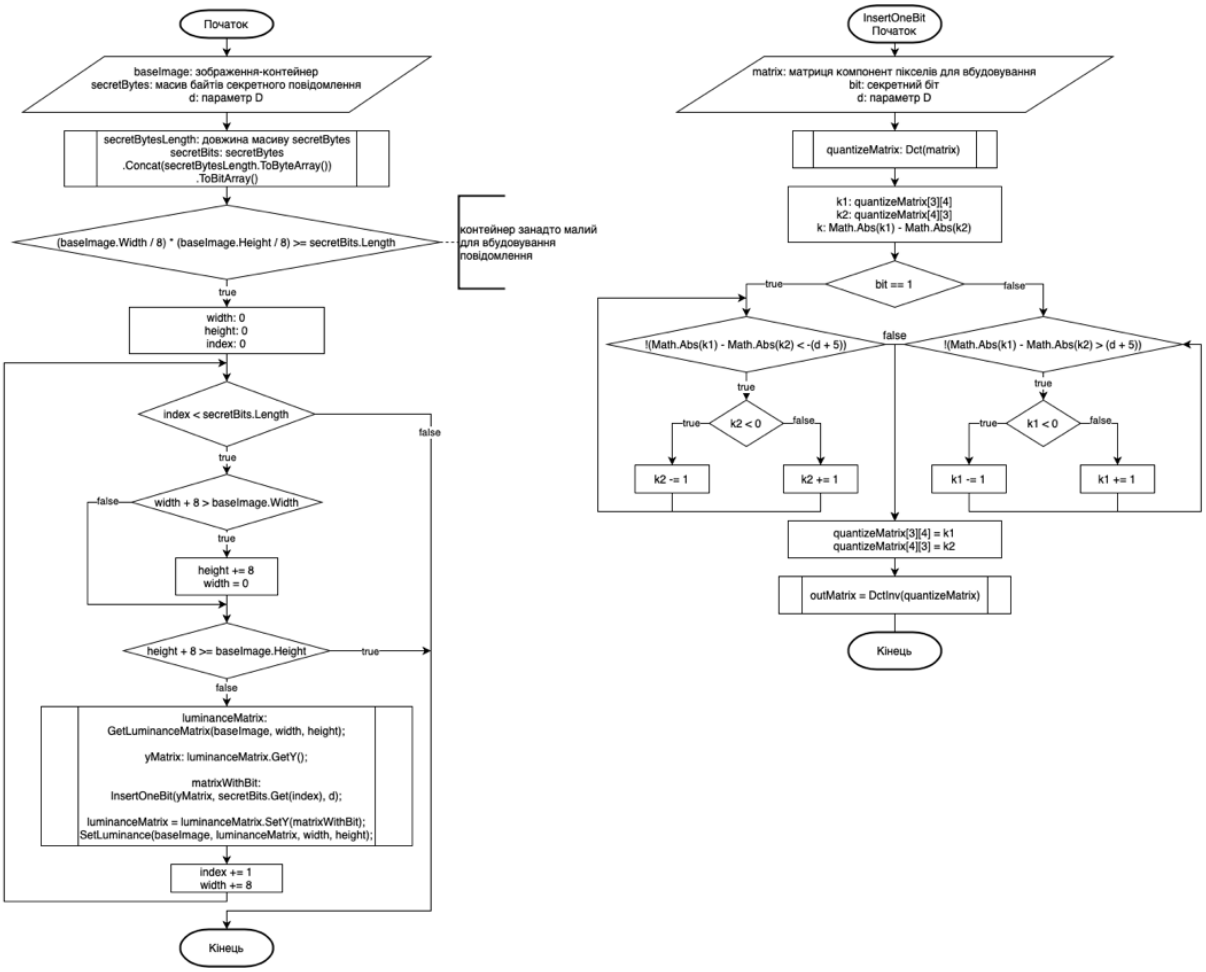
# ДОДАТОК Д

## Блок схема алгоритму витягнення методом НЗБ



# ДОДАТОК Ж

## Блок схема алгоритму вбудовування методом Коха і Жао



# ДОДАТОК И

## Блок схема алгоритму витягнення методом Коха і Жао