Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Стеганографічна система для приховування великих об'ємів інформації з обмеженим доступом
Назва теми

КвРКБ.180130.18.01.07 ПЗ
Шифр

Галузь знань *12 «Інформаційні технології»*
Шифр, назва

Спеціальність *125 «Кібербезпека»*
Шифр, назва

Освітня програма *«Кібербезпека»*
Назва

Виконав: студент IV курсу, група КБ–18–1 _____ Андрій МАТВІЙЧУК
Підпис Ініціали, прізвище

Керівник _____ Ігор МУЛЯР
Підпис, дата Ініціали, прізвище

Нормоконтролер _____ Сергій МОСТОВИЙ
Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри кібербезпеки та _____ Юрій КЛЬОЦ
Підпис Ініціали, прізвище

«8» червня 2022 р.

Хмельницький, 2022

№ рядка	Формат	Позначення	Найменування	Кількість	№ екземпляру	Примітка
1.	А4		Завдання на дипломний	1		
2.			проект			
3.	А4		Анотація	1		
4.			Стеганографічна система для			
5.	А4	КвРКБ.180130.18.01.07 ПЗ	приховування великих об'ємів			
6.			інформації з обмеженим доступом			
7.			Пояснювальна записка	1		
8.	А2	КвРКБ.180130.18.01.07 Е8	Структурна схема стеганосистеми	1		
9.						
10.	А2	КвРКБ.180130.18.01.07 Е8	Стеганоалгоритм	1		
11.						
12.						
13.	А2	КвРКБ.180130.18.01.07 Е8	Алгоритм оцінки	1		
14.			зміни зображення			
15.						
16.	А2	КвРКБ.180130.18.01.07 Е8	Результати тестування	1		
17.			додатку			
18.						
19.						
20.						
21.						
22.						
23.						
24.						
25.						
26.						

Зм.	Аркуш	№ докум.	Підп.	Дата	КвРКБ.180130.18.01.07 ВП		
Розроб.		Матвійчук А.В.			Стеганографічна система для приховування великих об'ємів інформації з обмеженим доступом Відомість проекту	Літера	Аркуш	Аркушів
Перевір.		Муляр І.В.				н	1	1
Н. Контр.		Мостовий С.В.		08.06.22		ХНУ, КБ-18-1		
Затверд.		Кльоц Ю.П.		8.0622				

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Освітній рівень бакалавр

Галузь знань 12 Інформаційні технології

Спеціальність 125 Кібербезпека

Освітня програма освітньо-професійна програма підготовки бакалаврів

ЗАТВЕРДЖУЮ

Завідувач кафедри кібербезпеки

01 03 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Матвійчуку Андрію Вікторовичу

Прізвище, ім'я, по батькові студента

1 Тема роботи Стеганографічна система для приховування великих об'ємів інформації з обмеженим доступом

Керівник роботи к.т.н, доц. Муляр І.В.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджено наказом ректора університету від 01.03.2022 № 18 додаток 10

2 Строк подання студентом роботи на кафедру: 3.06.2022

3 Вихідні дані до роботи криптографічні та стеганографічні засоби захисту інформації, формати Jpeg, BMP

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити) Дослідження предметної області та постановка задачі, аналіз стеганосистем та стеганоконтейнерів, розробка стеганоалгоритму з використанням форматних і просторових принципів приховування інформації, програмна реалізація та тестування засобу

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень) «Структурна схема стеганосистеми», «Блок-схема алгоритму оцінки змін фотографії із застосуванням матриці стандарту JPEG», «Стеганоалгоритм впровадження інформації», «Результати порівняння»,

6 Консультанти розділів курсового проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Мостовий С.В., старший викладач кафедри кібербзпеки	–	
Антиплагіат	Мостовий С.В., старший викладач кафедри кібербзпеки	–	

7 Дата видачі завдання _____ _____ 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір та затвердження теми кваліфікаційної роботи.	Січень	–
2	Аналіз об'єкта дослідження.	Січень–лютий	–
3	Проектування та розробка загальної архітектури і структури системи.	Лютий–березень	–
4	Програмна реалізація запропонованого рішення та тестування системи; аналіз результатів і оцінювання прийнятих рішень.	Квітень	–
5	Написання тексту пояснювальної записки та розробка графічних матеріалів.	Травень	–
6	Остаточне коригування кваліфікаційної роботи з урахуванням зауважень керівника.		–
7	Оформлення кваліфікаційної роботи як документа відповідно до вимог.		–
8	Отримання супровідних документів. Нормоконтроль.	Червень	–
9	Підготовка до захисту та захист кваліфікаційної роботи.		–

Студент _____ А.В. Матвійчук
Підпис Ініціали, прізвище

Керівник проекту (роботи) _____ І.В. Муляр
Підпис Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: Стеганографічна система для приховування великих об’ємів інформації з обмеженим доступом

Автор роботи: Андрій МАТВІЙЧУК

Керівник роботи: к.т.н., доц. Ігор МУЛЯР

Пояснювальна записка: 67 с., 23 рис., 2 табл., 2 дод., 24 джерела.

Ключові слова: стеганографічні методи, растрове зображення, захист інформації, прихований канал зв'язку.

Метою кваліфікаційної роботи є розробка стеганографічного алгоритму, та програмного застосунку для приховування великих об’ємів даних у файлах JPEG.

Запропонований алгоритм, дозволяє використовувати просторову зону файлу зображення та враховано можливість його функціонування в умовах втрати бітів під час міжформатних перетворень та стиснення. Розроблений застосунок використовується для приховування інформації з обмеженим доступом шляхом надсилання зображень в соціальних мережах, поштових клієнтах.

Дата 01.06.2022 Підпис студента

ЗМІСТ

					КвРКБ.180130.18.01.07 ПЗ		
Зм.	Арк.	№докум.	Підпис	Дата			
Виконав		Матвійчук А.В.		01.06	Стеганографічна система для приховування великих об'ємів інформації з обмеженим доступом Пояснювальна записка	Літера	Аркуш 2 / Аркушів 57
Перевір.		Муляр І.В.					
Н.контр.		Мостовий С.В.		01.06.23		ХНУ, КБ-18-2	
Затвер.		Кльоц Ю.П.		2.06.23			

ВСТУП

Із розвитком Інтернету та доступністю обчислювальних ресурсів для кожного «цифрову власність» можна миттєво відтворювати та розповсюджувати без втрати якості, фактично, безкоштовно. До цих пір інтелектуальна власність і цінність завжди асоціювалися з певним фізичним контейнером, який не можна було легко скопіювати, таким чином гарантуючи, що творець отримує вигоду від роботи.

Мережеві мультимедійні системи швидко розвиваються та розширюються, тому все більше інформації передається в цифровому вигляді; Розширення буде ще зростати, коли будуть розширені мультимедійні послуги, такі як електронна комерція, оплата за перегляд, відео за запитом, електронні газети, телеобробка, телеконсультації тощо. стають широко доступними. Однак автори, видавці та постачальники медіа-даних неохоче розповсюджують свої документи в Інтернеті, оскільки простота перехоплення, копіювання та перерозповсюдження електронних даних у точній оригінальній формі спонукає до порушення авторських прав. Тому для майбутнього розвитку мережевих мультимедійних систем важливо, щоб вони забезпечували методи захисту прав інтелектуальної власності власників даних від несанкціонованого копіювання та перерозповсюдження матеріалів, опублікованих у мережі. Класичні системи шифрування повністю не вирішують проблему несанкціонованого копіювання, оскільки після видалення шифрування документа немає контролю за його розповсюдженням.

Використання цифрових форматів відео та зображень швидко зростає з розвитком потокових медіа, онлайн-баз даних та електронних видань. Ця еволюція надає багато переваг, таких як просте, швидке та недороге копіювання продуктів. Однак це також збільшує потенціал для несанкціонованого поширення такої інформації та значно посилює проблеми, пов’язані із захистом авторських прав.

Досвід роботи систем критичної інфраструктури в умовах фальсифікації та активної протидії противника виявив гостру потребу в нестачі заходів щодо гарантування рівня безпеки інформаційних ресурсів. На деякий час це диктується важливим значенням інформаційних ресурсів критичної інфраструктури для процесів прийняття рішень, у тому числі в кризових ситуаціях. З багатьох інших причин підвищуються загрози конфіденційності та цілісності інформаційних ресурсів є. У значній мірі це обумовлено оперативних можливостей, програмного забезпечення та нформаційно-технологічних можливостей ворогуючої сторони. Тому безпека інформаційних ресурсів в інформаційних системах є важливою сферою прикладних досліджень.

Цікавим є розробка нових форм гарантування безпеки інформаційних ресурсів. Однією з областей дослідження є стеганографічні методи включення інформації в контейнер зображення. Основою для реалізації даного відео є системи відеоконференц-зв'язку, розгалужене мультимедійне обладнання, розробка відео-інформаційної кампанії, наявність офіційної інформації для конкретного відеоматеріалу та графічного матеріалу.

Враховуючи процес розвитку комп’ютеризованої стеганографії, можна стверджувати, що найближчим часом інтерес до розвитку її методів буде невпинно зростати. Актуальність проблеми захисту інформації постійно зростає і стимулює пошук нових методів інформаційної безпеки. В той же час, швидкий розвиток інформаційних технологій дає змогу реалізувати ці нові методи захисту.

Стеганографічні методи захисту інформації поряд із криптографічними займають важливе місце в інформаційній безпеці.

Серед стеганографічних методів окремий інтерес представляють методи безпосереднього вбудовування прихованої інформації в зображення-контейнер.

Характерною ознакою поширених технічних прийомів є те, що до об’єкта інформативного і нешкідливого змісту включається додаткове повідомлення з прихованою інформацією, яке потім відкрито передається до одержувача по каналу зв'язку.

Однак більшість відомих стеганографічних алгоритмів дозволяють приховати невеликі обсяги інформації [23]. Але на практиці досить часто виникає необхідність в прихованій передачі великих об’ємів інформації. Тому дослідження методів та засобів, що приховують великі об’єми інформації у поширених графічних форматах, є актуальною задачею.

Відповідно до мети написання кваліфікаційної роботи бакалавра передбачається розв’язання деякої актуальної практичної задачі в галузі кібербезпеки, і отримання відповідного прикладного результату у вигляді функціонально-придатного програмного засобу.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області і виявлення наявних проблем і завдань

Стеганографія – це метод організації спілкування, який фактично приховує саме існування комунікації [8]. На відміну від криптографії, де зловмисник може точно визначити, чи передане повідомлення є зашифрованим текстом, методи стеганографії дозволяють вбудовувати конфіденційні повідомлення в нешкідливі повідомлення, так що неможливо запідозрити існування вбудованого секретного повідомлення.

Необхідно розглянути аспекти, що визначають актуальність і трансцендентність стеганографічних методів в умовах кризи [11]. Тут ми повинні виділити наступні аспекти актуальності та значення стеганографічних методів:

а) необхідність підвищення рівня цілісності, конфіденційності, та доступності інформаційних ресурсів систем. У сучасних умовах функціонування систем критичної інфраструктури є необхідною умовою є підтримання заданого рівня складових інформаційної безпеки;

б) обмеження на використання криптографічних алгоритмів для захисту інформаційних ресурсів кризового призначення. Вони мають негативні наслідки та можуть завдати шкоди політичному та економічному іміджу держави;

в) формування умов для розробки стеганографічних підходів до забезпечення безпеки інформаційних ресурсів кризового призначення визначається такими позиціями:

- наявність великої кількості різноманітних стеганографічних методів прихованого вбудовування та передачі інформації;

- розвиток телекомунікаційних технологій з використанням відкритих широкосмугових каналів передачі даних;

- відсутність достатніх методів та засобів стеганографічного аналізу для виявлення фактів прихованого вбудовування спеціальної інформації;

- широке розповсюдження мультимедійних файлів в інтернет просторі. Це створює основу для формування контейнерів, які використовуються для вбудовування інформації;

- д) відсутність у нормативній базі обмежень щодо використання стеганографічних методів захисту інформації.

Тому комплексні системи захисту інформаційних ресурсів критичної інфраструктури також повинні використовувати методи стеганографічних перетворень. Стеганографічні методи, на відміну від криптографічної обробки, дають можливість приховати сам факт наявності секретного повідомлення. Тут інформація у вигляді повідомлення певним чином конвертується і вбудовується в якийсь цифровий контейнер, який не привертає уваги [21].

Узагальнена структурна схема стеганосистеми як системи передачі інформації наведена на рис. 1.1 [14].

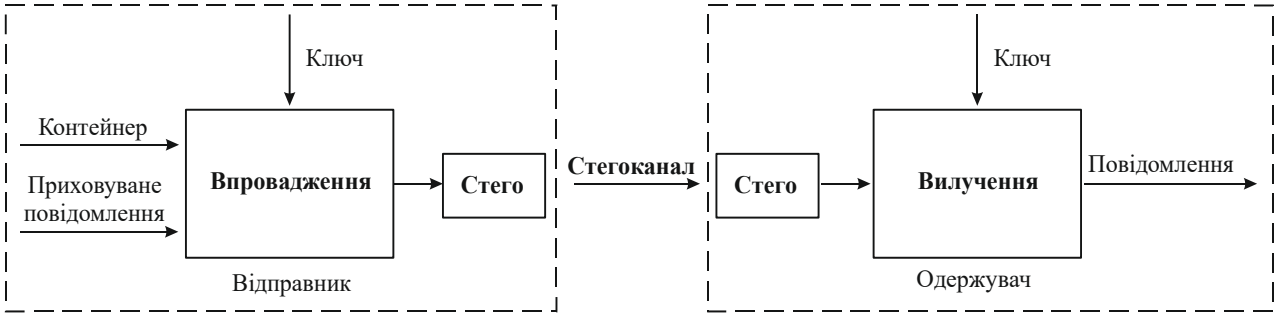


Рисунок 1.1- Структурна модель стеганосистеми

Відповідно до неї, на стороні відправника приховане повідомлення вбудовується в цифровий контейнер за спеціальним алгоритмом і ключем реалізації. Цей цифровий контейнер передається одержувачу через відкриті канали передачі інформації. На стороні одержувача вихідне повідомлення витягується з повного контейнера відповідно до алгоритму декодування та ключа пошуку.

Розповсюдження мультимедійних технологій сприяло розробці нових і вдосконаленню існуючих методів приховування інформації, а також сприяло створенню та реалізації більш досконалих методів та засобів організації прихованих каналів зв'язку, заснованих на особливостях представлення інформації в комп'ютерних файлах, пристроях, мережі інтернет, тощо.

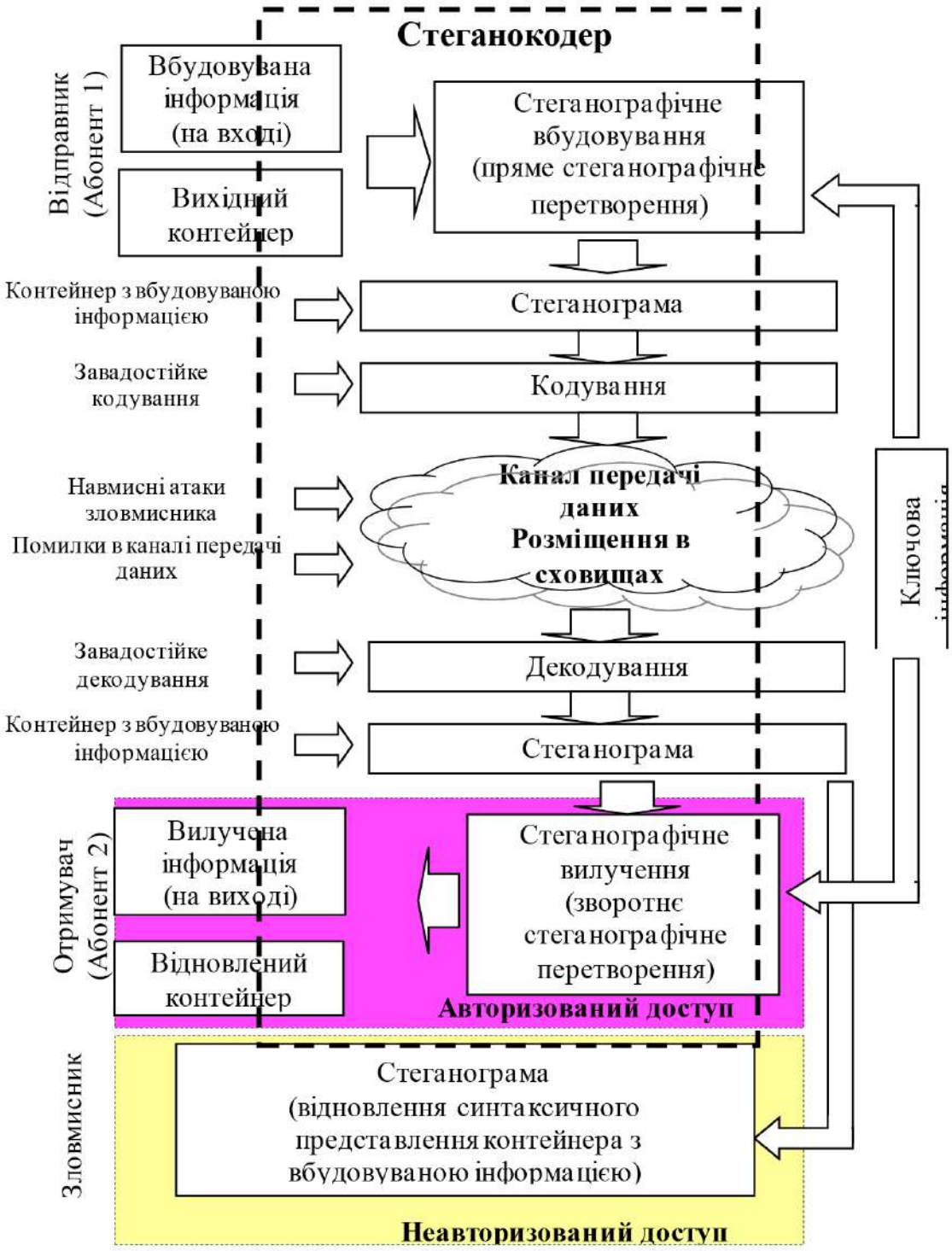


Рисунок 1.2 – Функціональна схема стеганографічної передачі інформації

Функціональна схема реалізації передачі латентних даних на основі використання стеганографічних підходів представлена на рисунку 1.2, і передбачає наступні кроки [23]:

1. Стеганографічне вбудовування. На цьому кроці здійснюється стеганографічне вбудовування інформації в пристосований цифровий контейнер. Вбудоване повідомлення можна попередньо конвертувати на основі шумостійких алгоритмів кодування, стиснення та шифрування. У стеганографічному кодері підготовлене перетворене повідомлення вбудовується в контейнер на основі певного стеганографічного правила та ключової інформації і формується стеганограма.

2. Пересилання стеганографічно перетвореного контейнера (стеганограми) одержувачу через канали передачі даних або для зберігання стеганограми. У процесі передачі інформації стеганограма може зазнавати активних та пасивних дій (атак).

3. Стеганографічне вилучення. На цьому кроці авторизований користувач виконує стеганографічне декодування з отриманої стеганограми. При цьому йому відома така інформація:

- факт наявності інформації, закладеної в стеганограму;
- правило для декодування;
- ключова інформація.

В результаті зворотнього стеганографічного перетворення авторизований користувач може отримати вбудовану інформацію.

Процес отримання вбудованої інформації здійснюється за наявності на приймальній стороні ключової інформації.

У випадку неавторизованого користувача у зловмисника немає інформації про наявність таємних вкладень повідомлення в конкретну стеганограму. Навіть якщо зловмисник знає, що в цій стеганограмі є дані, він не може вилучити їх через відсутність ключової інформації та правил декодування.

1.2 Огляд та класифікація стеганографічних методів

На сьогодні методи комп'ютерної стеганографії розвиваються за такими двома основними напрямками [25]:

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

– методи, засновані на надмірності при кодуванні аудіо та візуальної інформації.

Перший напрям заснований на використанні особливих властивостей форматів представлення комп'ютерних даних, а не на надмірності самої інформації. Спеціальні особливості форматів вибираються для захисту прихованого повідомлення від того, щоб його можна було почути, побачити чи прочитати безпосередньо.

На сьогодні стеганографічні методи дуже популярні серед шкідливих і шпигунських програм. Інструменти контролю зловмисного програмного забезпечення, включаючи захист периметра, можуть дуже мало зробити з агрегатами, заповненими корисним навантаженням. Такий носій дуже важко виявити, оскільки він виглядає як звичайні файли зображень (або файли інших типів). Усі поточні програми виявлення стеганографії, по суті, засновані на тестах, і їхня логіка не може бути реалізована в комерційних інструментах безпеки, оскільки вони повільні, мають відносно низький рівень виявлення, а іноді навіть містять математичні помилки.

Проаналізувавши на цьому етапі існуючі методи прихованої передачі інформації, можна запропонувати новий підхід до класифікації методів комп’ютеризованої стеганографії [24]. Доповнюючи та систематизуючи існуючі методи, ми можемо згрупувати їх за ознаками: призначення, вибір контейнера, за наявністю ключа та способами приховування даних (рис. 1.3).

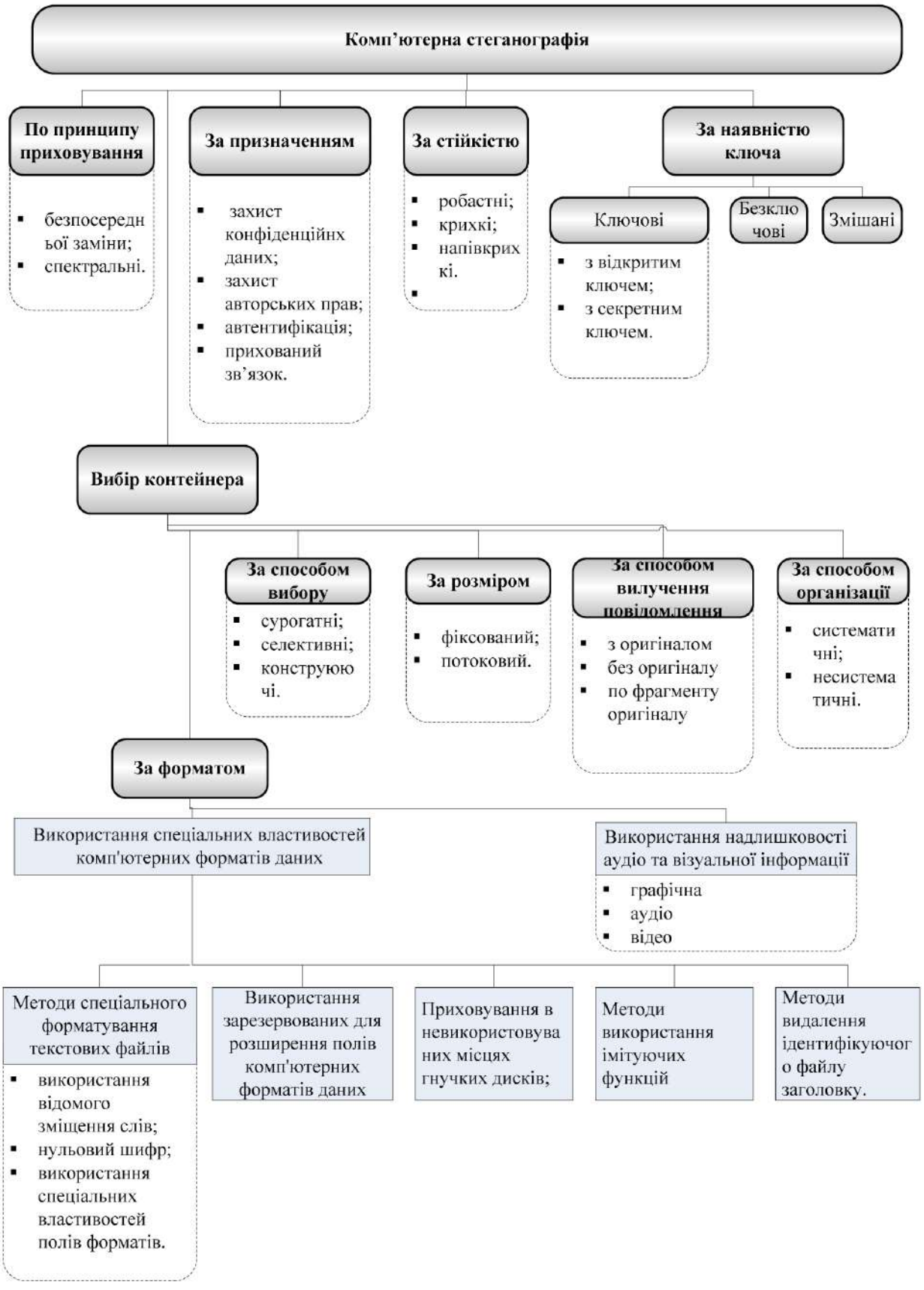


Рисунок 1.3 –Класифікації стеганографічних засобів

За призначенням існуючі стеганографічні методи можна розділити за такими сферами використання:

1. Захист від копіювання (авторське право, електронна комерція, розповсюдження мультимедійної інформації тощо);

2. Прихована анотація електронних документів (мультимедійні бази даних, медичні зображення, картографія);

3. Аутентифікація (голосова пошта, конфіденційний електронний бізнес, системи відеоспостереження

4. Приховане спілкування (може використовуватися для передачі інформації в цілях військової розвідки, злочинних угрупувань, а також у випадках, коли нормативно-правовими актами використання криптографії заборонено).

1.3 Методи приховування інформації в графічних файлах

Набір стеганографічних методів можна розділити на два класи на основі принципів, які використовуються при їх побудові, а саме а саме не форматні та неформатні Неформатні методи неминуче призводять до спотворення зображення, викликаного тим, що прихована інформація безпосередньо використовує основні дані зображення. Зауважимо, що цей клас стеганографічних методів досить стійкий до активних і пасивних атак.

Форматні методи засновані на властивостях конкретного формату зображення, що використовується в потоці. Для їх реалізації проводиться детальний аналіз формату зображення, його структури, щоб знайти певні обслуговуючі біти, зміни, які не будуть втрачені і не спотворять вихідне зображення. Однак всі форматні методи, з урахуванням принципу загальновідомості стеганографічної системи, мають загальний недолік - для них не складно побудувати автоматичний алгоритм, спрямований на виявлення факту приховування іформації. Тому вони мають низьку стійкість до пасивних

атак.

Методи просторового приховування засновані на принципі заміни бітів зображення, що містять зайву і незначну інформацію, бітами секретного повідомлення.

Найпоширеніший і основний метод (на якому базується більшість цієї групи) полягає в заміні найменших значущих бітів (LSB) послідовних пікселів у зображенні бітами секретної інформації. Зазвичай обсяг прихованої інформації менше обсягу вихідного зображення, тому після успішного застосування стеганографічного методу отримують дві області з різними статистичними властивостями, які можна ідентифікувати методами статистичного стеганалізу. Щоб уникнути розкриття факту приховування, вбудоване повідомлення доповнюється інформаційним шумом - випадковими бітами. Кількість такого інформаційного шуму вибирається таким чином, щоб його довжина в бітах дорівнювала кількості пікселів на зображенні. Розглянемо переваги використання цього методу: безсумнівну простоту виконання і значну корисну місткість контейнера. Недоліком є те, що при будь-якому спотворенні контейнера вбудована інформація також буде спотворена.

До найчастіше вживаних підходів приховування в просторовій області графічного зображення відносяться методи [15]:

- заміни найменш значущого біта;
- псевдовипадкового інтервалу;
- псевдовипадкової перестановки;
- заміни палітри;
- блочного приховування;
- Дамстедгера-Делейгла-Квіксвотера-Мака;
- квантування зображення;
- Куттера-Джордана-Боссена.

Розглянемо для прикладу декілька з них.

Метод заміни найменшого значущого біта (LSB) включає алгоритми, які дозволяють вам постійно замінювати біти ваших компонентів у кожному пікселі.

Піксель може бути представлений трьома числами (модель RGB), і кожен з цих компонентів можна змінити так, щоб його значення майже не змінювалося. Зазвичай вставляється один-три біти повідомлення в піксель.

На рисунку 1.4 наведено декомпозицію пікселя на складові в палітрі RGB (три байти) та позначено найменш значущі біти, які будуть замінені на біти прихованого конфіденційного повідомлення.

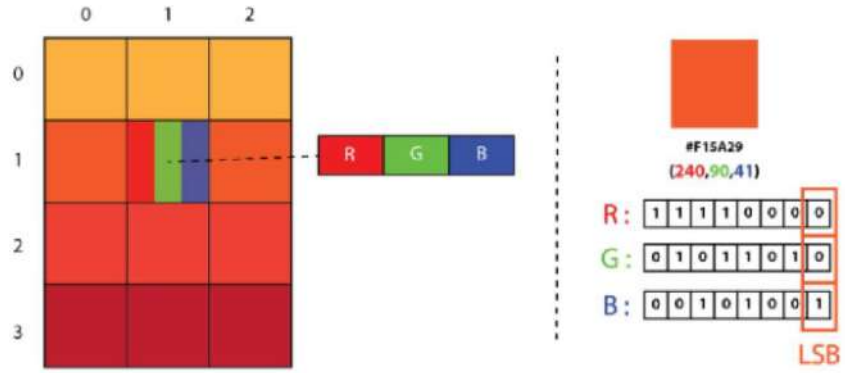


Рисунок 1.4 – Метод LSB

Недоліком LSB є його низька стеганографічна стійкість – будь-які спотворення контейнера можуть призвести до часткової або повної втрати повідомлення [7]. Потрібно зазначити, що при перегляді зображення з великим масштабом можуть бути чітко видно області з вбудованим повідомленням. Цей метод є достатньо популярним беручи до уваги простоту в реалізації.

Метод псевдовипадкових інтервалів застосовує технологію вбудовування повідомлення в один піксель, але за допомогою генератора псевдовипадкових чисел здійснюється вибір відстані між двома пікселями виконується [11]. Перевага цього методу полягає в тому, що при невеликих повідомленнях відносно обсягу контейнера набагато складніше візуально визначити наявність зашифрованої інформації. Недоліком залишається низька стійкість до стеганалізу, оскільки лише відстань між вбудованими бітами може здаватися випадковим.

Метод псевдовипадкової перестановки покращує спосіб розташування

цільових пікселів так, що їх порядок також вибирається псевдовипадковим чином [13]. Псевдовипадковий генератор будує послідовність індексів пікселів, які додатково вказують, у який піксель потрібно вставити фрагмент конфіденційного повідомлення. Ці алгоритми можуть гарантувати рівномірний розподіл секретних бітів у просторі контейнера.

Метод блоків заснований на принципі поділу контейнера на блоки довільного розміру, кожен з яких обчислює біт парності найменших значущих бітів [12]. Якщо біт парності не дорівнює секретному біту, то інвертування одного з молодших бітів у блоці змінить його значення на потрібне.

Метод заміни палітри використовує порядок кольорів в палітрі пікселів зображення для вбудовування повідомлення. Так як розташування кольорів в палітрі не важлива, то можна здійснити M! їх перестановок, тому існує можливість вбудовування невеликого секретного повідомлення. Цей метод вважається нестійким.

В підході Дамстедгера-Делейгла-Квіксвотера-Мака попередньо відбувається пошук найкращих блоків 8x8 пікселів для вбудовування повідомлення. Відповідно до результатів проведеного пошуку відповідний біт повідомлення інтегрується в кожен піксель блоку. Цей метод є стійким до JPEG-стиснення, так як розмір блоку відповідає компресійним алгоритмам, та володіє підвищеною стійкість за рахунок надлишковості кодування.

Метод Каттера-Джордана-Боссена вбудовує фрагменти секретного повідомлення, змінюючи компонент яскравості або синього кольору, роблячи деформацію контейнера невидимою для людського ока.

Алгоритм має стабільність до багатьох відомих методів стеганалізу, але пошук та видобування повідомлення може бути невдалим, оскільки функції вбудовування та вилучення не є симетричними

BlindHide (приховування всліпу). В цьому підході біти повідомлення приховані в пікселях один за одним зліва направо, починаючи з першого рядка. Недоліком вважається той випадок, коли контейнер заповнений не повністю і, отже, верхня частина зображення буде візуально зашумлена [3].

HideSeek (схованка). Для вибору порядку приховування бітів повідомлення використовується генератор псевдовипадкових чисел. Його перевага над алгоритмом сліпого приховування полягає в тому, що контейнер завжди буде рівномірно заповнений, але наявність вбудованого секретного повідомлення можна визначити візуально, оскільки воно не враховує характеристики контейнера [16].

Методи приховування в просторовій області нестабільні щодо стиснення з втратами, яке найчастіше застосовується в графічних форматах, що спричиняє спотворення вбудованої інформації. Це істотний недолік при використанні таких методів. Однак це не стосується методів приховування у частотній області.

Для представлення зображення в частотній області використовується кілька основ, з яких зображення декомпозується. Серед найбільш використовуваних є дискретне перетворення Фур'є (ДПФ), дискретно-косинусне перетворення (ДКП), вейвлет-перетворення, метод головних векторів (перетворення Карунена-Лоева). Ці перетворення застосовуються до окремих частин зображення або до зображення в цілому.

Одним з базових методів приховування секретної інформації в частотній області зображення є алгоритм Коха- Жао. Суть цього мктоду полягає в тому, що зображення розбивається на кілька блоків 8х8 пікселів, з цих блоків вибирається масив виділених піксельних компонентів, цей масив піддається дискретному косинусному перетворенню, підстановці вибраних коефіцієнтів і зворотному перетворенню. Принцип вбудовування полягає в тому, щоб змінити коефіцієнти ДКТ так, щоб їх різниця по модулю була більшою за додатній параметр алгоритму (для вбудовування «0») або меншою від'ємного параметру алгоритму (для вбудовування «1»). По суті, алгоритм Коха-Жао є алгоритмом вбудовування прихованого повідомлення в процес стиснення JPEG (рис. 1.5).

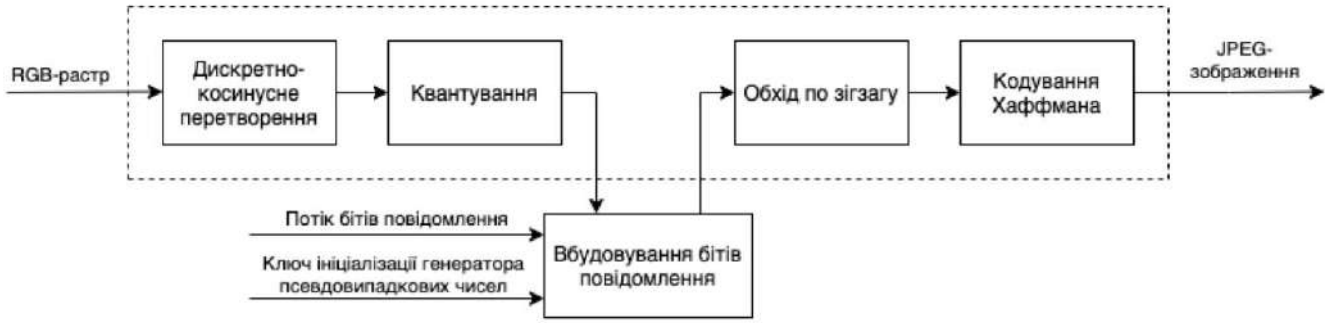


Рисунок 1.5 – Алгоритм Коха-Жао

Основним недоліком алгоритму Коха-Жао є висока складність розрахунку ДКП.

Алгоритм Бенгама-Мемона-Ео-Юнга оптимізує метод Коха-Жао, вибираючи блоки так, щоб їх зміни були мінімальними, а для вбудовування використовується ще один фактор ДКП, завдяки чому зображення менше візуально спотворюється.

Є дві вимоги при виборі блоків для для вбудовування. Блоки повинні бути вільні від різких переходів яскравості і повинні бути досить однотонними.

Метод *Xy-By* був створений для вбудовування цифрового водяного знаку шляхом зміни коефіцієнтів ДКП блоків у вибраному контейнері. Практикується створення зображення цифрового водяного знаку в чорно-білих тонах. Перед вбудовуванням вся матриця ділиться на 255, а після вилучення множиться на це ж число.

Коли використовується метод Фрідріха, зображення перетворюється в сигнал з ненульовим математичним очікуванням і певним стандартним відхиленням. Після проведених маніпуляцій розраховані низькочастотні коефіцієнти потраплять в певний діапазон і значення по модулю низькочастотного коефіцієнта дискретного косинусного перетворення вхідного сигналу не перевищить певного рівня. Тому модифікації підлягають лише низькочастотні коефіцієнти ДКП.

1.4 Постановка задачі

У першому розділі на основі аналізу вітчизняної та зарубіжної літератури, патентів і наукових статей представлено поняття стеганографії а також класифікація методів та завдань прихованої передачі конфіденційної інформації.

Проведено аналіз інтернет-ресурсів у перспективних напрямках, у яких стеганографія може бути використана як інструмент захисту інформації в системах критичної інфраструктури, в комплексних системах захисту інформації, автоматизованих системах обробки даних, що дозволило перейти до структурної та функціональної схеми стеганографічної системи.

На основі вивчення відомих публікацій досліджено надійність та стабільності стеганографічної системи за типами атак проти неї.. Наведено результати існуючих інформаційно-теоретичних досліджень з проблеми приховування інформації у разі активної контратаки з боку зловмисника. В результаті було закладено міцний теоретичний фундамент для розробки засобів стеганографічного приховування конфіденційної інформації. Аналіз показав, що область трансформації орієнтована на введення невеликих обсягів даних, які є невеликою послідовністю байтів.

Метою дипломного проєкту є розробка стеганографічних алгоритмів, що дозволяють приховано вбудовувати великі об’єми інформації безпосередньо у графічні файли найбільш поширених форматів.

Для досягнення цієї мети в бакалаврській роботі необхідно вирішити такі завдання:

1. Проаналізувати класи стеганографічних алгоритмів.

2. Дослідити застосування різних підходів у розробці алгоритмів приховування інформації.

3. Розробити алгоритм, який дозволив би використовувати просторову область зображення, при цьому врахував можливість спотворення стиснення і дозволив би вставляти багато секретної інформації в зображення

стеганоконтейнера.

5. Передбачити надійну роботу розробленого стеганоалгоритму в умовах втрати бітів під час операції перетворень і стиснення між графічними форматами.

6. На основі запропонованого алгоритму розробити програмне забезпечення для ОС Windows.

2 ДОСЛІДЖЕННЯ ТА АНАЛІЗ СТЕГАНОСИСТЕМ ТА СТЕГАНОКОНТЕЙНЕРІВ

2.1 Критерії якості стеганосистеми

Три основні завдання захисту інформації полягають у забезпеченні конфіденційності, цілісності та доступності інформації. Ці три властивості інформації утворюють тріаду CIA (confidentiality, integrity, availability) [19]. Оскільки стеганографію можна віднести до ключових елементів інформаційної безпеки, як і криптографія, конфігурація апаратного забезпечення, інженерні та організаційні завдання, вона також повинна відповідати глобальним завданням інформаційної безпеки та забезпечувати вимоги CIA. Звичайно, все залежить від мети, у разі приховування факту передачі інформації стеганографія повинна забезпечити конфіденційність і цілісність вкладеної секретної інформації.

Обсяг секретної інформації, яку необхідно приховати, і стійкість цих даних до перешкод і шуму вимагають реалізації різних методів приховування. Це пов’язано з тим, що кожен із методів приховування даних спрямований на певну мету і не може досягти всіх задач стеганографії відразу. Отже, виникає потреба в ефективному методі порівняння існуючих алгоритмів та вибору найбільш оптимального для конкретного випадку. Перш за все, необхідно визначити основу мети стеганографії, а вже потім на їх основі формування відповідні критерії відбору. Таким чином, крім досягнення мети CIA щодо інформаційної безпеки, до стеганографії додаються додаткові задачі, утворюючи для неї спеціальну тріаду, яка представлена на рис. 2.1, робастність (robustness), місткість (capacity), непомітність (imperceptibility) запропоновані Wang [10].

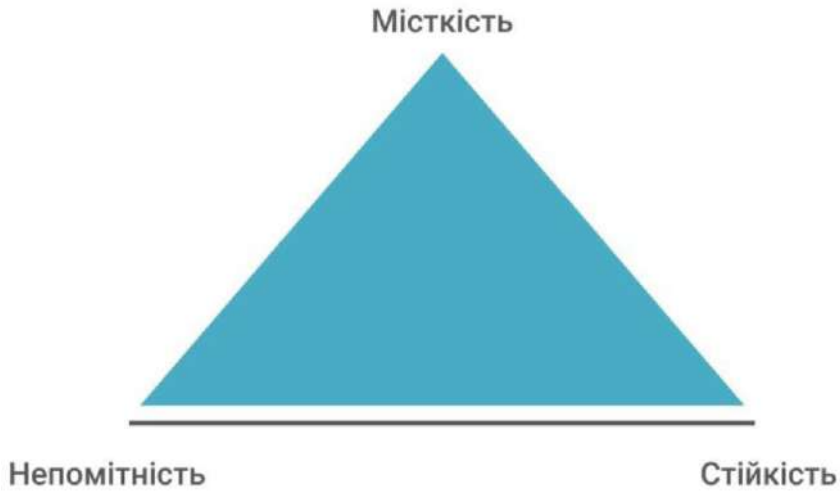


Рисунок 2.1 – Стеганографічна тріада RCI

Стійкість (робастність) відноситься до здатності отримувати конфіденційну інформацію від стегоконтейнера після навмисного спотворення контейнера, переданого через канал зв'язку, з завадами. Атаки стеганалізу використовуються для визначення стабільності стеганографічних методів,.

Місткість — це максимальна кількість інформації, яку можна вставити в стеганографічний контейнер. В першу чергу ця функція залежить від стеганографічного алгоритму та властивостей контейнера. Метрики, які використовуються для вимірювання місткості: кількість бітів конфіденційного повідомлення, які можуть бути вбудовані в піксель зображення; співвідношення між розміром конфіденційного повідомлення та максимальним розміром повідомлення, яке може бути інтегровано в цей контейнер [12].

Зі збільшенням обсягу повідомлення розмір стеганоконтейнера, файлу, який виконує роль контейнера, викличе певні підозри. Таким чином, існує залежність надійності приховування від обсягу повідомлення, що показано на графіку (рис. 2.2).

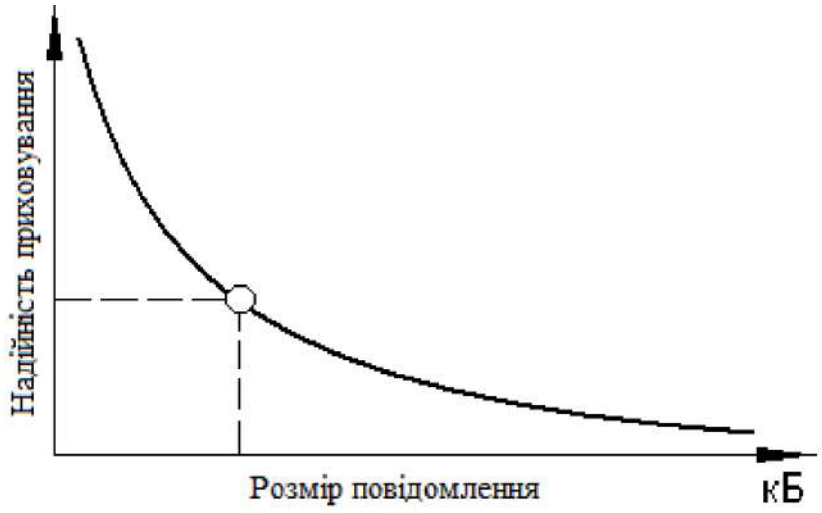


Рисунок 2.2 –Надійність приховування даних від розміру конфіденційного повідомлення

Змінюючи певні якості стеганоконтейнера, можна досягти високої надійності приховування повідомлення або його великого об’єму, але важко досягти обох показників одночасно, оскільки зростання одного з цих значень безпосередньо призводить до зниження в іншому.

Властивість невидимості надається, коли між стеганоконтейнером і оригінальним контейнером немає істотної різниці або вона досить мінімальна, тобто візуально «невидима» для зловмисника [13]. Для математичної оцінки вимоги непомітності застосовують такі показники: максимальна та середня різниця, кореляція, відношення сигнал/шум.

Основне завдання стеганалізу — виявити наявність секретного повідомлення [22]. Загалом, існуючі методи стеганалізу не дають конкретної відповіді на те, чи є повідомлення вбудоване в зображення чи ні, а лише вказують на деяку ймовірність його існування в отриманому повідомленні.

Існує два види стеганалізу цифрових зображень. До першої групи відносяться направлені методи, які використовуються для пошуку вбудованого повідомлення у випадку фактичного використаного стеганографічного алгоритму. До другої групи відносяться «сліпі методи», які застосовуються за відсутності знань про раніше використаний стеганографічний алгоритм.

Найбільше поширення отримали статистичні методи стеганалізу. Серед них:, метод оцінки частоти бітових послідовностей у матриці, що містить елемент, метод оцінки кількості переходів найменш значущих бітових значень у сусідніх пікселях, аналіз гістограм, аналіз розподілу пікселів у площині, перевірка розподілу елементів для значень монотонності, розподіл на основі критерію X^2.

Порівняльний аналіз — це загальнонауковий метод пошуку та виявлення схожості чи відхилень від однотипних властивостей досліджуваних об’єктів на основі емпіричного дослідження або зібраних статистичних даних.

Порівняльний аналіз використовується для вивчення систем і об'єктів технічного, символічного, органічного, соціального характеру. Основною умовою застосування методу є наявність у досліджуваних об’єктів хоча б однієї загальної властивості, в межах якої можна виділити варіації досліджуваних змінних.

Існує два базових типи порівняльного аналізу:

- з'ясувати суттєві характеристики двох або більше споріднених об'єктів шляхом порівняння їх подібних властивостей;

- встановити схеми розвитку одного і того ж досліджуваного об’єкта, порівнявши його стани та властивості в різні періоди.

Виконання порівняльного аналізу включає наступні етапи:

- формулювання критеріїв порівняння;

- визначення об'єктів та одиниць аналізу;

- перевірка методологічної еквівалентності порівняння;

- оцінка характеристик обраних об'єктів;

- інтерпретація спільних і відмінних властивостей.

Алгоритми порівняльного аналізу визначаються метою конкретного дослідження. Залежно від того, як порівнюються об’єкти, у статичному чи динамічному аналізі. Порівняльний аналіз є досить ефективним інструментом для перевірки гіпотез і побудови теорій, оскільки він допомагає відокремити загальні характеристики та фактори від унікальних.

Результати порівняльного аналізу є основою для подальшого застосування алгоритму прийняття рішень.

За рівнем секретності стеганосистеми поділяють на практично стійку, теоретично стійку, та нестабільну [22].

Теоретично стійкі стеганосистеми приховують інформацію в тих частинах контейнеру, значення елементів яких не перевищує рівень шуму або певних помилок, і теоретично було показано, що неможливо створити стеганоаналітичний метод для виявлення прихованої інформації.

Практично стабільні стеганосистеми здійснюють таку модифікацію фрагментів повідомлення, що можливе виявлення прихованої інформації, але відомо, що злочинець не має таких стеганоаналітичних методів.

Нестабільні стеганосистеми модифікують контейнер таким чином, що існуючі стеганоаналітичні системи можуть розкривати конфіденційну інформацію. У цьому випадку стеганоаналіз дозволяє виявити слабкі сторони зазначеної системи для її подальшої модифікації до теоретично стійкої або, практично стабільної стеганосистеми.

Стеганографічна система вважається порушеною, якщо порушник зміг довести наявність прихованого повідомлення в перехопленому контейнері. Допускається, що злочинець може використовувати будь-який тип атаки та має необмежені обчислювальні можливості. За аналогією з криптоаналізом виділяють наступні типи атак на стеганосистеми [14]:

- на базі існуючого відомого заповненого контейнеру;
- на базі вибраного прихованого повідомлення;
- на базі відомого вбудованого повідомлення;
- на базі певного заповненого контейнеру;
- на базі відомого порожнього контейнера (цей метод не має аналога в криптоаналізі);
- на базі вибраного порожнього контейнера (цей метод не має аналога в криптоаналізі);

- на базі відомої математичної моделі контейнера або його частини (цей метод не має аналога в криптоаналізі).

2.2 Особливості графічних файлів, як стеганографічних контейнерів

JPEG - це один з популярних і досить потужних алгоритмів. Він працює з блоками 8x8, де колір і яскравість змінюються досить плавно. В результаті, коли двовимірний масив такого блоку розкладається на подвійний рядок уздовж косинусів, вартими уваги є лише перші коефіцієнти. Тому, стиснення в JPEG проходить за рахунок плавної зміни кольорів існуючого зображення.

Структуру файлу, формату JPEG зображено на рисунку 2.3 [17].

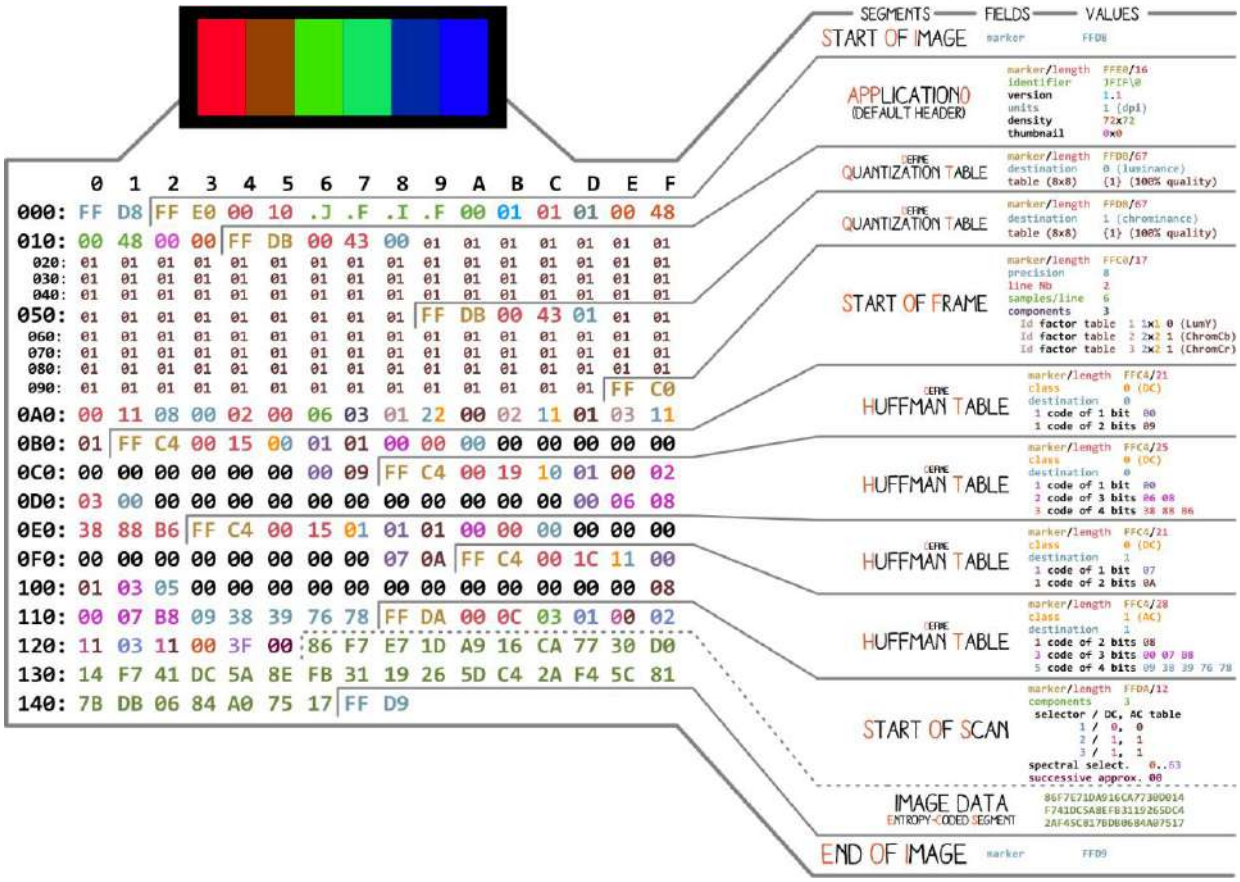


Рисунок 2.3 – Структура файлу, формату Jpeg

В основі цього формату лежить універсальна структура:

− Title (Заголовок 2 байти) : $ff, $d8 (SOI);

− Частина APPO;

− Деяка довільна кількість "фрагментів" (подібні IFF (Image File Format) частинам);

− Кінець (End 2 байти) : $ff, $d9 (EOI) [25].

Усі блоки мають таку структуру;

− Title (Заголовок 4 байти) ;

− $ff ідентифікатор;

− n клас, розміром 1 байт;

− sh, sl – розмір;

− довжина фрагменту, максимально 65533 байти.

В структурі двійкового файлу міститься кілька маркерів (або заголовків). Їх можна сприймати, як своєрідні закладки. Вони необхідні для роботи з файлами і використовуються такими програмами, як переглядач в різних ОС, щоб ми могли дізнатись деталі зображення. Маркери вказують, на те, де саме зберігається певна інформація у файлі. Найчастіше маркери розташовують відповідно до значення довжини певного сегмента.

Першим важливим маркером у структурі файлу є FF D8. Він маркує початок зображення. Якщо ми не зможемо знайти його, то можна припустити, що маркер розташований в якомусь іншому файлі. Також важливим є маркер FF D9. Він, визначає що отримано кінець файлу зображення. Після кожного маркера, не враховуючи діапазон FFD0-FFD9 та FF01, далі слідує значення довжини послідовності цього маркера. А стандартні маркери початку та кінця файлу JPEG завжди мають довжину по два байти.

Відповідно до специфікації JPEG:

- графічні зображення, закодовані тільки одним компонентом, є градацією сірого, де 0 - це чорний, а 255 - білий.
- графічні зображення, що визначаються трьома характеристиками, вважаються RGB-даними, представленими в просторі YCbCr. Якщо зображення

містить фрагмент маркера APP14, то - це колірне кодування RGB або YCbCr відповідно до даних в фрагменті маркера APP14. Відношення між RGB та YCbCr визначається відповідно до вимог стандарту ITU-T T.871 | ISO / IEC 10918-5.

- графічні зображення, які кодуються чотирма компонентами, вважаються CMYK -Відповідно (0,0,0,0) означає білий колір. Якщо зображення містить фрагмент маркера APP14, то відповідне колірне кодування вважається CMYK або YCCK залежно від інформації в сегменті маркера APP14.

В переважній більшості реалізацій алгоритму JPEG замість колірної моделі RGB використовуються значення яскравості (luminance) і хроматичності (chrominance) (кодування YUV). Це актуально, тому що людське має здатність погано розрізняти високочастотні зміни в яскравості на маленьких відрізках, тому можна зменшити частоту, але людина не помітить різниці.

- як і в моделі RGB кожен піксель кодується трьома байтами кольорів (червоного, зеленого і синього), так і в моделі YUV також використовується три байти, але вони мають інше призначення: компонента Y визначає яскравість кольору (luminance, або luma), параметри U і V визначають колір (chroma), компонета U відповідає за долю синього кольору, а компонета V - за частку червоного.

Алгоритм був запропонований групою експертів у галузі фотографії, спеціалізований для стиснення 24-бітових зображень. Взагалі, в основі алгоритму лежить дискретно косинусне перетворення (ДКП), застосоване до масиву пікселів зображення, щоб отримати деяку нову матрицю коефіцієнтів. Відповідно зворотнє перетворення використовується для безпосереднього отримання вихідного зображення. ДКП поводить декомпозицію зображення на амплітуди певних частот. Тому, при перетворенні ми отримуємо масив, в якому більшість коефіцієнтів або близькі, або нульові. Враховуючи недосконаліть людського зору, можна певним чином оптимізувати коефіцієнти без помітної втрати якості зображення.

Оскільки множення великих масивів займають багато часу, досить важко проаналізувати одразу все зображення. Тому кожне зображення поділяють на

невеликі масисви з елементами 8x8, і далі визначають частотні компоненти в відповідних блоках зображення, зменшуючи при цьому кількість цих блоків, зберігаючи лише значущі, при цьому записують їх більш економічним способом..

Метод стиснення JPEG складається з наступних етапів:

а) заміна палітри на кольорову схему YCbCr;

б) піддискретизація кольорових складових шляхом усереднення груп пікселів;

в) застосування дискретного косинусного перетворення для переходувід просторового до частотного подання;

г) квантування;

д) етап "зигзагоподібого" сканування;

д) первинна компресія;

д) вторинне стиснення [16].

Розглянемо базові концепції і методики кодування, які використовуються в JPEG. А процес декодування буде виконуватися в зворотному порядку.

Етапи JPEG – компресії зображені рисунку 2.4.

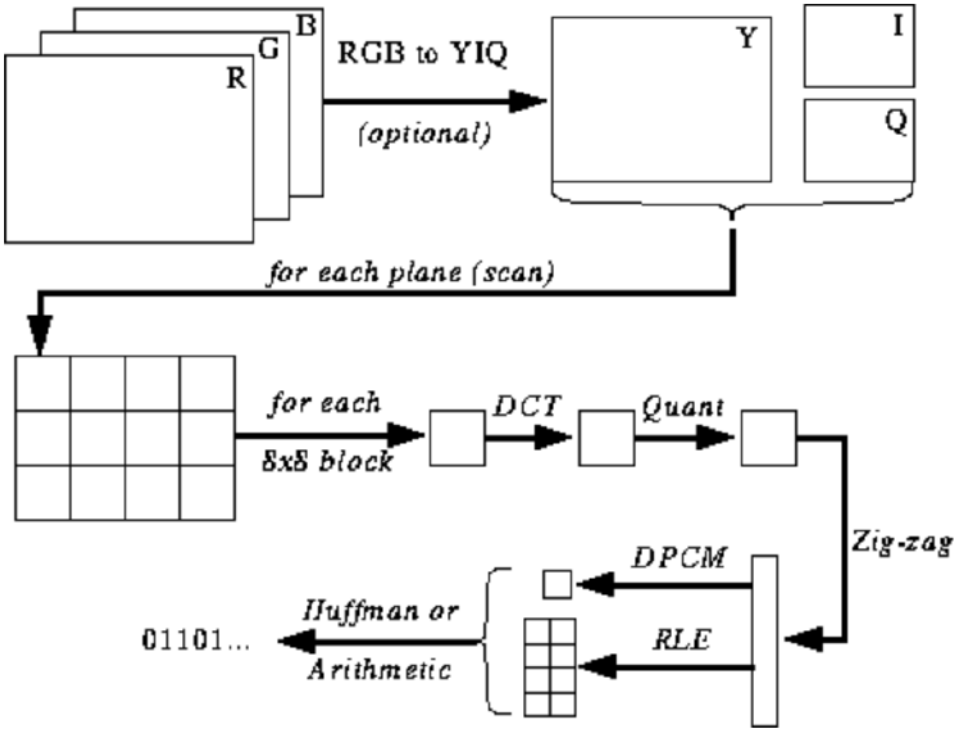


Рисунок 2.4 – Етапи JPEG – кодування

Стандарт JPEG визначає ряд таблиць кодів Хаффмана для здійснення опису різних способів шифрування отриманих квантованих даних.

Квантовані коефіцієнти ДКТ мають, як правило, два різні значення параметрів.

Перший - це коефіцієнт, який визначає перше значення в упорядкованій зигзаг матриці. Це значення завжди є середнім значенням у межах дозволеного діапазону.

Друге - коефіцієнт від другого до останнього в упорядкованій зигзаг матриці. Значення коефіцієнтів близькі до нуля, при цьому імовірність того, що вони будуть нульовими, зростає разом зі збільшенням коефіцієнтів біля кінця матриці.

Якщо дані каналу не представляють цілу кількість блоків, кодер повинен заповнити решту неповних блоків деякими видами фіктивних даних. Ви можете заповнити межі фіксованим кольором (наприклад, чорним), який звучать артефактами вздовж видимої частини межі; Повторення крайових пікселів є поширеною технікою, яка зменшує (але не обов'язково повністю усуває) такі артефакти, і можна використовувати більш складні методи заповнення країв.

Ці два типи інформації досить відрізняються. Це виправдовує їх поділ та застосовувати різні методи кодування для отримання оптимальної стратегії та ефективності стиснення.

Крім того, в методі кодування JPEG також розрізняється, чи є необхідна інформація даними про яскравість. Для цього є можливість вказувати різні таблиці коду для кожного з блоків, як визначаються таблиці квантування.

JPEG перетворює зображення в матрицю 8x8 пікселів (називаються MCU, Minimum Coding Unit, тобто мінімальні одиниці кодування), та змінює діапазон значень пікселів таким чином, щоб в центрі матриці було значення 0, потім застосовує до кожного блоку ДКП і стискає результат за допомогою процесу квантування (рис. 2.5).

Рисунок 2.5 – Алгоритм стиснення Jpeg

Метод стиснення даних JPEG, що дозволяє стискати окремі зображення, можна розділити на три етапи [21]:

Крок 1 - відбір кольорової інформації та субдискретизація;

Крок 2 - блок дискретних косинусних перетворень;

Крок 3 - квантування та кодування отриманих дискретних значень перетворенних косинусів.

Перший етап - це субдискретизація кольорової інформації.

Кожен піксель вхідного зображення, представлений 3 байтами в системі RGB, трансформується в модель YUV (яскравість, насиченість, тон) відповідно до перетворень:

$$Y=77/256\cdot R+150/256\cdot G+29/256\cdot B$$
$$U=131/256\cdot R-110/256\cdot G-21/256\cdot B+128$$
$$V=-44/256\cdot R-87/256\cdot G+131/256\cdot B+128$$

або в матричному представлені:

$$\begin{pmatrix} Y \\ U-128 \\ V-128 \end{pmatrix} = \begin{pmatrix} 0{,}301 & 0{,}586 & 0{,}113 \\ 0{,}512 & -0{,}430 & -0{,}082 \\ -0{,}172 & -0{,}340 & 0{,}512 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Трансформація з моделі *YUV* в систему *RGB* виконується за наступними формулами:

$$R=Y+1{,}37\cdot(U-128)$$
$$G=Y-0{,}698\cdot(U-128)-0{,}336\cdot(V-128)$$
$$B=Y+1{,}73\cdot(V-128)$$

або в матричному представлені:

$$\begin{pmatrix} R+175{,}4 \\ G-132{,}4 \\ B+221{,}4 \end{pmatrix} = \begin{pmatrix} 1 & 1{,}370 & 0 \\ 1 & -0{,}698 & -0{,}336 \\ 1 & 0 & 1{,}730 \end{pmatrix} \begin{pmatrix} Y \\ U \\ V \end{pmatrix}$$

Далі значення компоненти *Y* залишаються без зміни, а характеристики значень компоненти *U* і *V* зменшується (субдискретизація). Можливі різні алгоритми субдискретизації: заміна значень сусідніх пієселів зображення на їх середні значення. При цьому можливі кілька варіантів усереднення пікселів: дві по горизонталі, дві по вертикалі, або квадрат з чотирьох сусідніх точок. Самий

поширений варіант: число точок зменшується вдвоє, і значення точок обчислюються згідно з виразом:

$$y(n)=1/4\cdot x(n-2)+1/2\cdot x(n-1)+1/4\cdot x(n).$$

При цьому матриця розміру 8*16 значень компонент *U* або *V* перетворюється в матрицю 8*8 значень.

При декомпресії інформації для покращення якості проміжні пікселі рекомендується отримувати не простим повторенням, а шляхом інтерполяції між сусідніми пікселями. На практиці використовується наступний алгоритм: при відтворенні зображення матриця 8*8 точок перетворюється в матрицю 8*16 точок за формулою:

$$x(n)=[y(2n)+y(2n-1)]/2.$$

Якщо використовується описаний алгоритм розбиття, то на практиці досягається стиск зображення в 1,5 рази. Дійсно 1 байт яскравості пікселя залишається без змін, а кожні 2 байти сладових *U* та *V* заміняються на 1 байт. Тому, замість 6 байт на кожні 2 пікселя зображення тепер припадає тільки 4 байти.

При використанні варіант об'єднання чотирьох сусідніх пікселів можливий стиск зображення в 2 рази. Тому що 1 байт складової яскравості не змінюється, а кожні 4 байти складових *U* та *V* заміняються на 1 байт. Отже, замість 12 байт на кожні 4 пікселя зображення тепер вистачає 6 байт.

Мінімальний фрагмент інформації для опрацювання – це матриця початкового *RGB* зображення розміром 8*16 елементів. В результаті обробки такого фрагменту на першому кроці отримуємо чотири блоки розміром 8*8, також два блоки розміру 8*8 для характеристики яскравості *Y* та по одному блоку розміру 8X8 для характеристик *U* та *V*. Це зображено на рисунку 2.6.

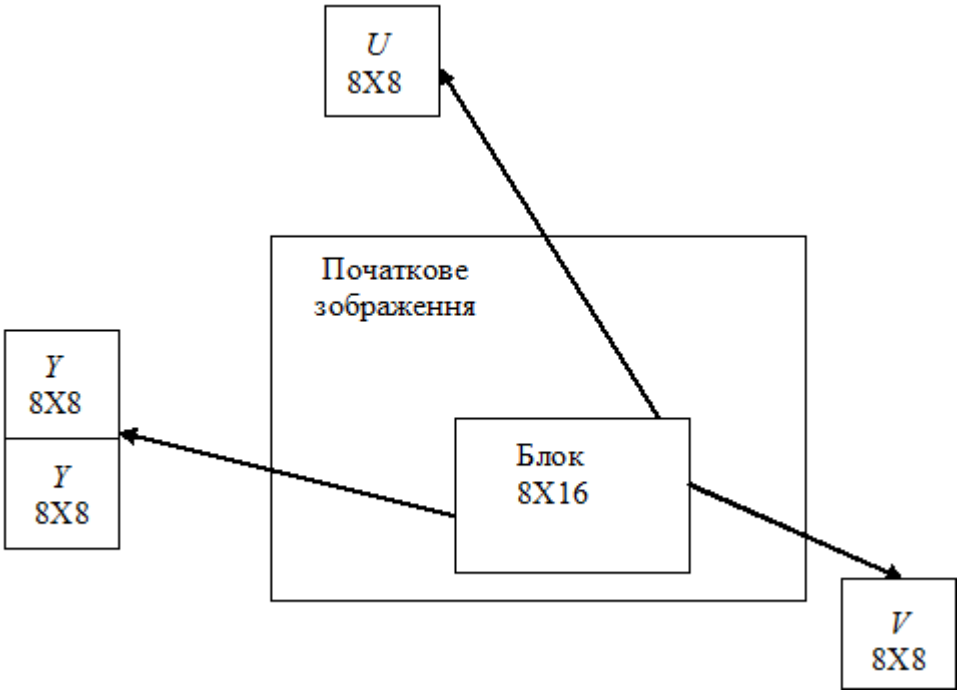


Рисунок 2.7 - Субдискретизація кольорової інформації

Другий етап стиску базується на ДКП. Перетворення зображення на набір косинусів на перший погляд виглядає марним, але ДКП приймає блок розміром 8x8 пікселів і інформує їх, як відтворити цей блок, використовуючи масив 8x8 косинусних функцій (рис 2.8):

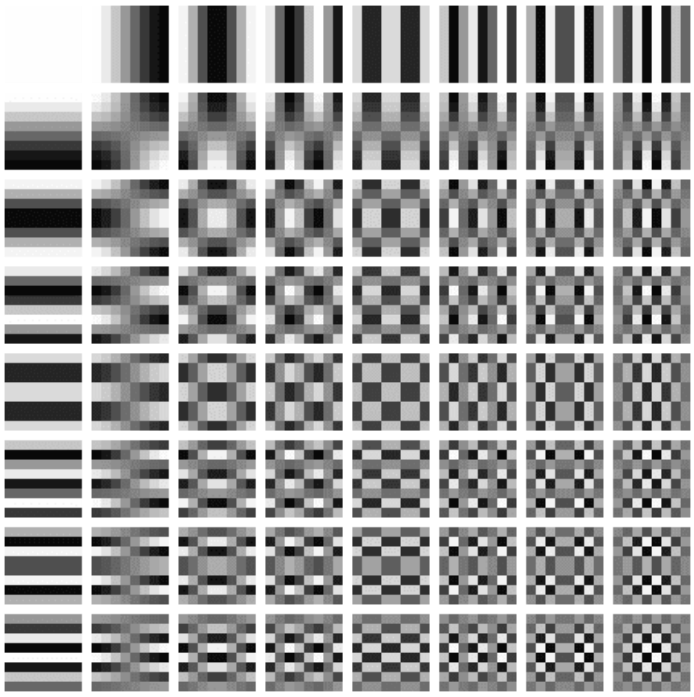


Рисунок 2.8 – Матриця ДКП

Ми застосовуємо алгоритм ДКП окремо до кожного піксельного компонента. Після цього отримуємо матрицю коефіцієнтів 8x8, яка показує частку кожної (з усіх 64) отриманих косинусних функцій у вхідну матрицю 8x8. Властивість матриці коефіцієнтів ДКП полягає в тому, найбільші значення, як правило, знаходяться у верхньому лівому куті матриці, а найменші - в правому нижньому. У верхньому лівому куті матриці знаходиться функція косинуса найнижчої частоти блоку, а в правому нижньому куті - найвища частота пікселя блоку.

Це відбувається тому, що більшість зображень мають величезну долю низькочастотної інформації та невелику частку високочастотної. При умові, що нижньому правому компоненту кожного масиву ДКП присвоїти значення 0, то отримане зображення для людського виглядати однаково, оскільки людина не спроможна легко розрізнити високочастотні зміни. Саме в цьому полягає наступний крок.

Кожен із елементів Y, U, V зображення на цьому етапі мприймається як окреме монохромне зображення та його стиснення виконується окремо.

Зображення поділяється на окремі блоки з елементами 8X8.

$$P_{ДКП}=APA^T,$$

де A - масив двовимірного ДКП,

$P_{ДКП}$ - масив значень ДКП фрагменту зображення.

Двовимірне ДКП має ту особливість, що воно зосереджує найбільші за величиною по модулю значення переважно у верхньому лівому куті масиву значень перетворення. Типовий розподіл коефіцієнтів ДКП відображено в наведеній далі матриці:

1347,4	168,4	−15,5	−35,7	34,1	20,8	23,6	0
223,9	−0,4	−126,1	99,3	−14,5	18,2	0,5	−0,2
48,4	−134,8	89,4	−12,2	18,9	0,1	0,2	−0,1
−11,3	21,8	11,9	−0,8	24,5	−35,8	0,2	−0,2
0,1	0,5	−0,5	−0,1	−48,1	0,5	−0,8	−0,1
0,4	−0,1	0,5	0,2	1,1	−0,3	−0,1	−0,2
−0,4	−0,1	0,7	0	−0,4	0,1	0,8	0,5
−0,2	−0,2	0,4	−0,2	−0,4	−0,3	0,4	0

Ці значення потрібно брати з більшою точністю, а решта значень ДКП можна значно підсилити. Це лежить в основі стиснення графічного зображення за допомогою дискретних косинусних трансформацій. Методи, що реалізують описану подію, відбуваються на наступному третьому кроці.

На третьому кроці компресії ми виконуємо процес квантування та кодування значень дискретного косинусного перетворення.

На першому етапі виконується квантування значень ДКП. Для цього створюється матриця Q дільників з коефіцієнтами $q(i, j) = 1 + (1 + i + j) * r$, $i, j = 0,1,\ldots, 7$; *де* r - параметр, який визначає ступінь компресії і одночасно якість відтвореного зображення. Для параметру *Y* рекомендується взяти $r = 2$, а для параметрів *U, V* значення r може бути більшим. Значення $q(i, j)$ - це крок квантування, який змінюється залежно від положення в зображені. При переході від верхнього лівого кута блоку до нижнього правого кута крок квантування виконується грубе квантування, тобто крок збільшується. Такі значення кроку квантування відповідають властивості ДКП найбільшого значення по модулю, головним чином у верхньому лівому куті масиву значень перетворення.

При коефіцієнті r=2 матриця дільників має вигляд:

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

Значення $x(i,j)$ масиву $P_{ДКП}$ діляться на відповідні значення $q(i,j)$ масиву дільників і заокруглюються до найближчого цілого. Сам процес квантування можна описати наступним виразом: round ($x(i,j)/ q(i,j)$).

Після кроку квантування отримують велику кількість нулів. Тому для отримання найдовшої послідовності нулів доцільно використовувати наступну схему обходу отриманої матриці.

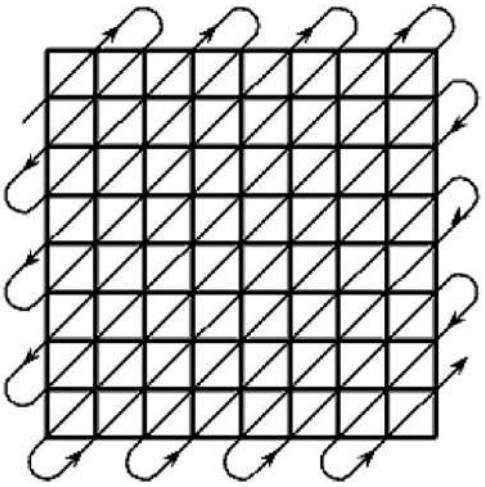


Рисунок 2.9 – Обхід матриці

Таким чином, на початку масиву формуються матричні коефіцієнти, що відповідають низьким частотам, а в його кінці - високі. Тому, в результаті обходу можна отримати послідовності до 30 нулів.

2.3 Висновки

Алгоритми, які використовуються до просторової області зображення, засновані на візуальній надлишковості сприйнятої інформації і тому в даний час не так популярні, як алгоритми області форматного перетворення. Це пов’язано з широким використанням формату JPEG, де колірні компоненти та складові якскравості пікселів приховані за зоною перетворення.

В порівнянні з JPEG- форматом, формат BMP має набагато ширшу просторову область зображення.

Формат JPEG містить великий набір компонентів файлової структури, які взагалі не мають впливу на просторову область зображення (тобто не спотворюють його) і деякі з них не обробляються програмним забезпеченням візуалізації JPEG-файлів.

Просторова область зображення найбільше підходить для введення великих обсягів інформації. Але при використанні формату JPEG ця область прихована за областю форматного перетворення та піддається стисненню. Область форматного перетворення у зв'язку з невеликою кількістю потенційних майданчиків для реалізації більше підходить для реалізації невеликого обсягу інформації, наприклад цифрових водяних знаків.

3 РОЗРОБКА СТЕГАНОАЛГОРИТМУ З ВИКОРИСТАННЯМ ФОРМАТНИХ І ПРОСТОРОВИХ ПРИНЦИПІВ ПРИХОВУВАННЯ ІНФОРМАЦІЇ

3.1 Алгоритм оцінки змін зображення

Стійкість зображення оцінюється за допомогою різних методів [21]. Один з них, це BER (Bit Error Rate) використання коефіцієнт помилкових бітів, який застосовується при оцінці модифікацій бітової послідовності зображення [18]:

$$BER(S,S'') = \frac{\sum p_i}{N},$$

де N – кількість біт зараження;

$p_i = 1$, при умові $s_i \neq s_i''$, або

$p_i = 0$, якщо $s_i = s_i''$,

s_i - j-й біт вхідного зображення;

s_i'' - j-й отриманого вихідного зображення.

Однією з популярних метрик при обчисленні рівня спотворень вмісту стеганоконтейнеру є *PSNR* (Peak Signal Noise Range), тобто максимум співвідношення «сигнал/шум», [12]:

$$PSNR = \frac{XY \cdot \max\limits_{x,y}\left(C_{x,y}\right)^2}{\sum\limits_{x,y}\left(C_{x,y} - S_{x,y}\right)^2},$$

де X, Y - розміри зображення;

$S_{x,y}$ - розмір пікселя модифікованого зображення;

$C_{x,y}$ - розмір пікселя вхідної картинки;

В ході виконання кваліфікаційної роботи було розроблено алгоритм оцінки змін зображення за допомогою масиву стандарту JPEG,

Підхід базується на дискретному косинусному перетворенні, яке застосовується до окремих блоків розміром 8x8 пікселів вхідного зображення. Матриця коефіцієнтів ДКП отриманого модифікованого зображення віднімається з матриці ДКП блоку вхідного зображення [20].

Початкове зображення приймається в якості сигналу, а за шум - зміни, що з'являються при модифікації.

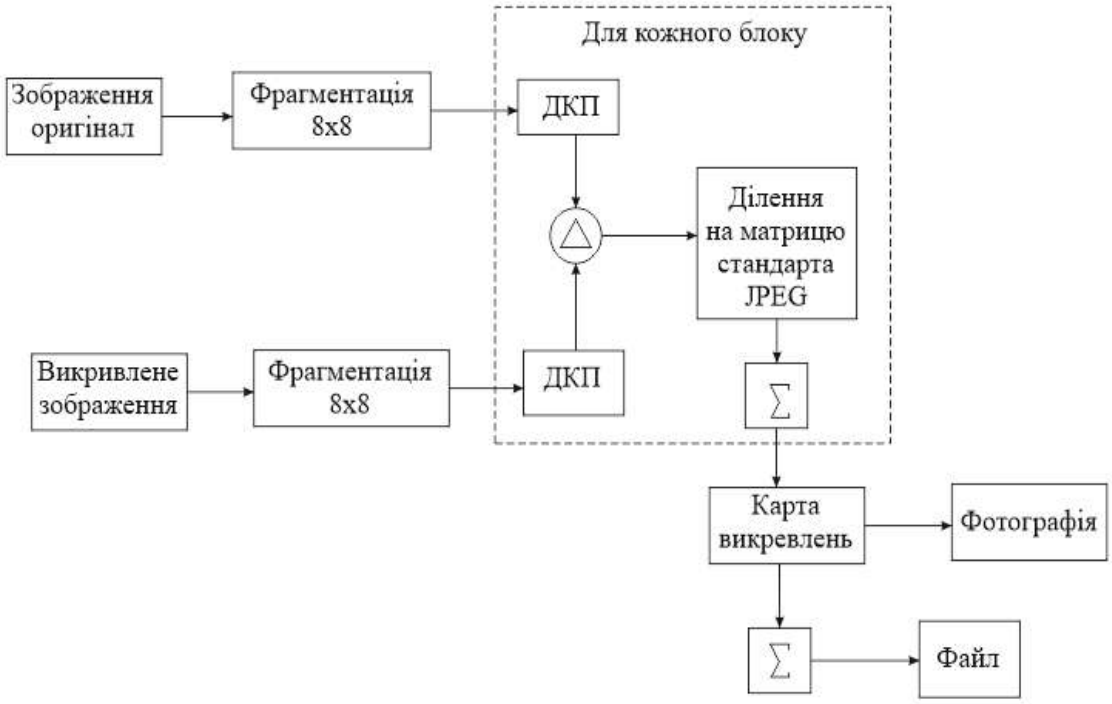


Рисунок 3.1 - Блок-схема алгоритму оцінки змін зображення

Отриманий двовимірний масив ділиться на матрицю квантуючих коефіцієнтів яка застосовується у форматі JPEG, зображену на рис. 3.2.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Рисунок 3.2 - Коефіцієнти квантування матриці

Розроблений метод порівняльного аналізу був використаний при розрахунку оцінки стійкості контейнеру до стискання з втратами JPEG..

Самі ж результати аналізу стійкості зображення до стискування з втратами JPEG зображені на рис. 3.3.

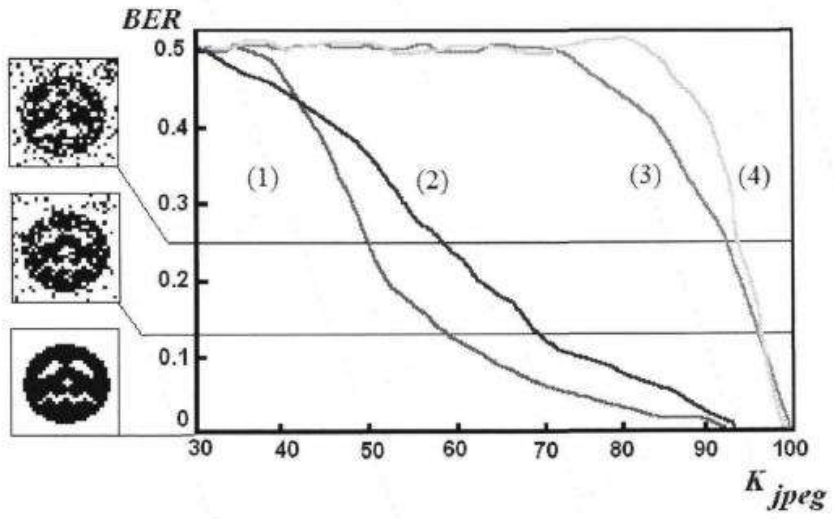


Рисунок 3.3 – Стійкість до стискання з втратами JPEG

3.2 Алгоритм впровадження прихованої інформації

Знання структури файлів JPEG та алгоритму стиснення з втратами є необхідною умовою для роботи з JPEG [16]. Візуалізація за допомогою засобів перегляду JPEG залежить від структури файлу формату JPEG. Зрозуміло, що введення в нього прихованої інформації не має призводити до візуального спостереження зміни зображення [13].

Отже, виходячи з попереднього аналізу, та особливостей структури формату JPEG, описаної вище, можна зробити висновок, що в цьому форматі існують маркери, які визначають сегменти, але на пряму не беруть участі у перетворенні JPEG. Таким чином вони не впливають на візуалізацію зображення, і не помітні людському зору. Перелічимо їх:

1. COM;
2. DAC;
4. DNL;
3. APP15;
5. маркери SOF2 - SOF10;
6. Невстановлені сегменти.

Файли BMP (BitMap) краще узгоджені з внутрішнім форматом Windows, який зберігає растрові дані в цьому форматі. Ім’я файлу зазвичай має розширення *. BMP, або іноді *.RLE. Це розширення означає кодування довжини послідовностей і вказує, що растрова інформація файлу була стиснута дійсним методом стиснення RLE, і ці методи також підходять для файлів BMP.

У файлах BMP колір будь-якого пікселя зображення представлений 1, 4, 8, 16 або 24 бітами (біт/піксель). Кількість біт/піксель, яка називається глибиною передачі кольору, визначає максимальний колір зображення. Зрозуміло, що зображення глибиною 1 біт/піксель може мати має лише 2 кольори. Файл BMP, що містить зображення в 256 кольорах (8 біт/глибина пікселя), завжди поділено на 4 основні частини: назва файлу растрової графіки; заголовок інформації

растрового масиву, таблицю кольорів та інформацію, що міститься в растровому масиві.

Така інформація, необхідна для візуалізації, як початкова адреса області даних растрового масиву, знаходиться в заголовку файлу формату BMP.

Інформаційний заголовок растрової таблиці містить конкретну інформацію про зображення, що зберігається у файлі, таку як висота або ширина зображення в пікселях.

Кольори RGB, які використовуються в кольоровому зображенні, зберігаються в таблиці кольорів.

Розроблений алгоритм орієнтований на такуструктур файлу BMP:

1. назва растрового графічного файлу (14 байт);

1.1. підпис файлу BMP (2 байти);

1.2. розмір файлу (4 байти);

1.3. не використвуються (2 байти);

1.4. не використовуються (2 байти);

1.5. адреса даних бітового масиву (4 байти);

2. заголовок інформації растрового масиву (40 байт);

2.1. довжина заголовка масиву (4 байти);

2.2. висота зображення (4 байти);

2.3. ширина зображення (4 байти);

2.4. кількість колірних площин (2 байти);

2.5. глибина біт/піксель (2 байти);

2.6. метод компресії (4 байти);

2.7. довжина растрового масиву (4 байти);

2.8. вертикальне розширення (4 байти);

2.9. горизонтальне розширення (4 байти);

2.10. число кольорів у зображенні (4 байти);

2.11. кількість використовуваних основних кольорів (4 байти);

3. дані растрового масиву (змінна величина)

Одні із особливостей, які виникають при збереженні зображення, полягає в тому, що воно зберігається в порядку знизу вгору. Обсяг пам'яті, виділений для зберігання будь-якого рядка, становить 4 байти. Невикористані комірки містять «шум». Найзначніший біт (тетрада) порівнює значення крайнього лівого пікселя. Під час збереження зображення True Color кожен піксель відповідає трьом послідовним байтам, які зберігають кольори компонентів B, G, R.

Розглянемо детальніше особливості цього підходу. Структура алгоритму запису прихованої інформації з обмеженим доступом зображено на рис. 3.4.

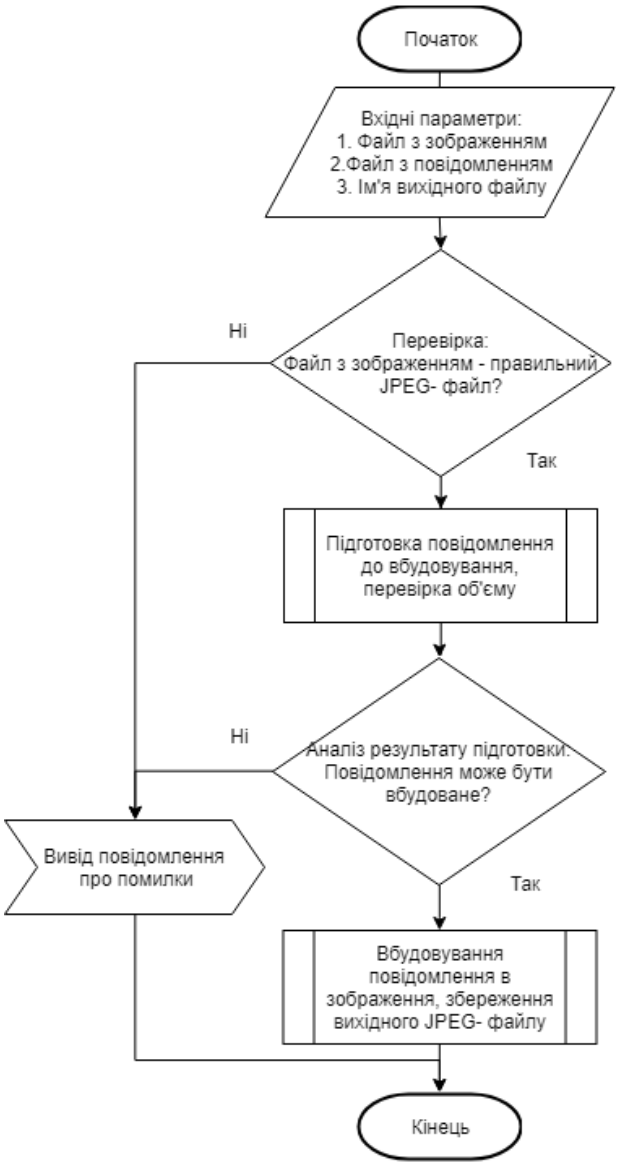


Рисунок 3.4 - Алгоритм запису прихованої інформації з обмеженим доступом

Запропонований алгоритм, використовуючи структуру формату для приховування інформації, використовує перелічені маркери. Сегменти, позначені переліченими маркерами, дозволяють записувати інформацію. Але необхідно враховувати обмежений розмір кожного сегменту, який задається в розмірі двох байт - OxFFFF.

За необхідності, конфіденційні дані повинні бути стиснені та зашифровані перед впровадженням. Тоді необхідно враховувати параметри реалізації та розмір контейнера.

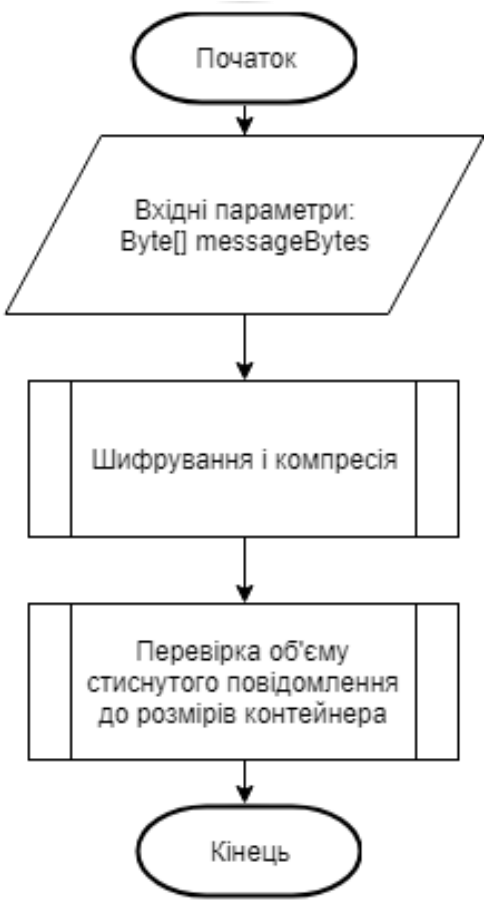


Рисунок 3.5 - Алгоритм підготовки інформації з обмеженим доступом до вбудовування

Оскільки в даному алгоритмі використовується формат JPEG, в якому вже на програмному рівні реалізовано стиснення з втратами, при впровадженні конфіденційної інформації з обмеженим доступом необхідно вжити ряд

попереджувальних заходів, щоб вберегти інформацію від спотворення при стиснені .

Вбудовування інформації з обмеженим доступом відбувається за алгоритмом, наведеним на рис. 3.6.

Рисунок 3.6 – Алгоритм вбудовування інформації з обмеженим доступом

Запропонований підхід можна описати наступним кроками. На першому етапі файл JPEG перекодовується у файл BMP. Це призводить до істотного збільшення розміру потоку. Враховуючи особливості BMP-формату, а саме, що для кодування одному пікселю ставиться у відповідність 3 байти (RGB), розмір

потоку суттєво збільшується, саме тому існує можливість додавання великого масиву прихованої інформації з обмеженим доступом.

Людське око ділить вхідний сигнал на окремі компоненти. Кожен елемент збуджує нервові закінчення людського ока через ряд підканалів. Відібрані оком елементи мають різні просторові та частотні характеристики, а також різну орієнтацію в просторі (горизонтальну, вертикальну, діагональну) [14]. Під час впливу на око двох компонентів з схожими характеристиками збуджуються однакові підканали. Саме тому додатній шум набагато помітніший на більш гладких ділянках зображення, ніж на високочастотних ділянках, тобто відбувається своєрідне маскування. Максимальний маскуючий ефект проявляється, коли два сигнали мають однакову орієнтацію в просторі та розташування.

Зорова система людини, згідно досліджень, має низьку чутливість до коливань блакитних відтінків. Саме тому запис прихованих даних часто відбувається за допомогою B-ї складової RGB-структур [17].

Але відомо, що наше око також погано контролює коливання найменш значущих розрядів складових R і G [15].

Найпростіший метод заміщення бітів зображення полягає в послідовній заміні в кожному b-байті.

Щоб мінімізувати зміни в просторовій області, реалізована стеганосистема за замовчуванням використовує лише молодший біт такого байта. Це дозволяє мінімізувати ймовірність виявлення навіть на зображеннях із великою синьою заливкою. Його наведено на рисунку 3.7.

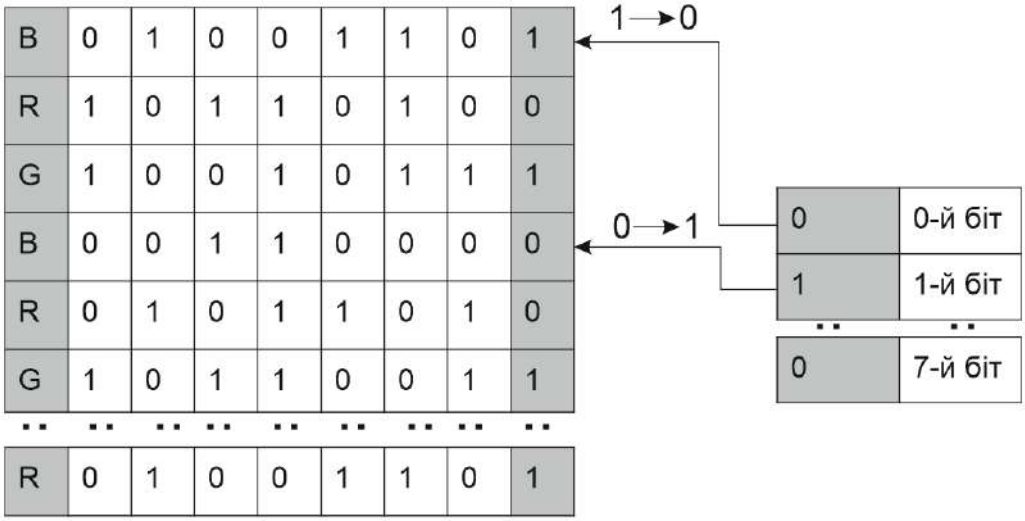


Рисунок 3.7 - Заміна бітів зображення

Зрозуміло, що розмір секретної прихованої інформації пропорційно залежить від кількості найменш значущих бітів. При використанні запропонованого алгоритму після стиснення JPEG внаслідок додавання прихованих конфіденційних даних, складова неприродного шуму зникає. Це дає можливість працювати як з одним, так і з чотирма значущими бітами блакитної компоненти пікселя.

Залежність потенційного розміру прихованої інформації можна вирахувати за формулою:

$$V = f(x,i) = \frac{x \overset{jpeg-bmptrans}{\rightarrow} x_{bmp} - H_{bmp}}{D_{bmp}} * i,$$

де x - розмір JPEG-файлу;

i є *[1;4]* – кількість використаних найменш значущих бітів;

H_{bmp} - розмір отриманого BMP- файлу, байт;

D_{bmp} - кількість бітів, які припадають на 1 піксель зображення в форматі BMP.

У таблиці 3.1 наведено дані розрахунків залежності розміру вбудованої інформації від інших параметрів.

Таблиця 3.1 - Залежність розміру вбудованої інформації з обмеженим доступом від розміру JPEG - контейнера і відповідного проміжного BMP-контейнера

№	Байтові послідовності	Файл 1	Файл 2	Файл 3
1	Зображення JPEG	297641	2561232	4339765
2	Проміжний BMP	2102702	21244718	40254950
3	Кількість біт синього каналу -1	87572	885735	1682604
4	Кількість біт синього каналу -2	175274	1779462	3355418
5	Кількість біт синього каналу -3	262757	2664209	5028122
6	Кількість біт синього каналу -4	350208	3548844	6691716

Як видно з таблиці, найбільший об'єм виходить перетворенням в 24 - бітове BMP- зображення, тому запропонована структура стеганосистеми базується на ньому.

Ключовим моментом даного алгоритму є виконання обов’язкове попереджувальних дій, щоб не було втрат прихованих секретних даних при стисненні.

Модифіковані молодші біти блакитної складової достаються з потоку та зберігаються в буфері 1. Потім в інший тимчасовий буфер, назвемо його 2 записується данні про вбудовування бітів переданої секретної прихованої інформації з обмеженим доступом (наприклад її порядок, підряд, або через один байт).

Далі модифікований растровий BMP файл піддається JPEG-кодуванню з найбільшим коефіцієнтом якості.

Потім відбувається етап отримання прихованих інформації з обмеженим доступом, що включає:

1. Копіювання існуючого JPEG- потоку в тимчасовий потік.

2. Перекодування та трансформація цього потоку в формат BMP.

3. Добування перекодованих байтів блакитної компоненти, з буферу 2.

В подальшому проходить дослідження витягнутої інформації на збіг з буфером, та вираховується різниця, яка записується в буфер 3.

Підготовча робота потрібна для подальшого одержання прихованих даних і вирішує проблему компенсації втрат. Інформація з буфера 3 дублюється в COM-сегмент файлу. Можна використати інший фрагмент, що регулюється програмним забезпеченням, наприклад DAC, DNL, SOF2 ÷ SOF10. Потім він доповнюється до вихідного JPEG- потоку. В подальшому цей потік записується у кінцевому файлі. Коефіцієнт якості трансформації BMP ->JPEG зазвичай менше |6|. Потім інформація буфера 2 також записується в вихідний файл в один з доступних невикористовуваних сегментів.

Також записується службова інформація:

1. Довжина прихованого повідомлення: (для відповідної синхронізації процесів приховування та отримання даних).
2. Прапор стиску: (вказує чи архівувалася конфіденційна інформація до приховування). По ній приймається рішення про необхідність проведення розархівування витягнутого байтового потоку
3. Прапор шифрування: (вказує чи шифрувалася інформація з обмеженим доступом до приховування)

3.3 Висновки

Розроблено алгоритм, який дозволяє збільшити корисний об’єм стеганоконтейнера що дає змогу приховування великих обсягів інформації з обмеженим доступом.

В цьому алгоритмі використовується подвійне перетворення формату. Інформація вбудовується в перетворений потік байтів BMP, а службові дані, що використовуються для вилучення, зберігаються як частина звичайної структури сегментів файлу. Ця концепція зробила алгоритм актуальним. Об’єм впровадженої інформації з обмеженим доступом обраховується по формулі :

$$V = f(x,i) = \frac{x \overset{jpeg-bmptrans}{\rightarrow} x_{bmp} - H_{bmp}}{D_{bmp}} * i$$

Вмикаючи режими 2/4, 3/4 і 4/4 найменш важливих бітів, запропонований алгоритм дозволяє приховати данні, об'єм яких можна вимірювати в мегабайтах.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЗАСОБУ

4.1 Вимоги до програмного додатку

Як було розглянуто в попередніх розділах, розроблений алгоритм орієнтований на вирішення проблеми прихованої передачі інформації з обмеженим доступом. Як і будь-який інший стеганографічний алгоритм, він може виконувати функції введення/виведення. Цією інформація можуть бути ідентифікаційні номери, цифрові водяні знаки та будь-яка конфіденційна інформація. Важливою і дуже корисною функцією є автоматизація функціональності реалізації/інтеграції та легкість та зручність програмної реалізації алгоритму для вибору вхідних/вихідних параметрів.

При впровадженні використовуються наступні параметри:

– файл, що містить контейнер;

– файл, що містить необхідну інформацію для впровадження (можливо зашифровану);

– ім’я вихідного файлу;

– стеганоключ (при необхідності).

Вихідними параметрами є файл із стеганоконтейнером.

Необхідні параметри для екстракції:

– файл з контейнером заповненим конфіденціною інформацією;

– ім’я вихідного файлу;

– стеганоключ (при необхідності).

Вихідними параметрами для екстракції є файл із отриманим повідомленням.

Далі показано результати дослідження за етапами програмної реалізації описаного алгоритму і їх застосування при вирішенні окремих завдань (рис. 4.1)

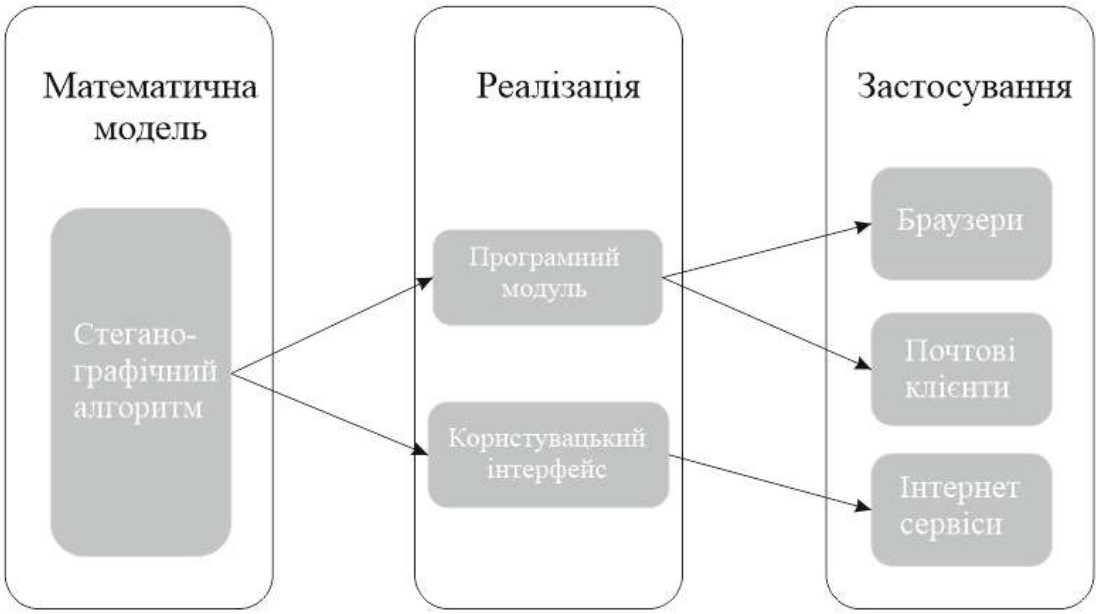


Рисунок 4.1 – Етапи створення ПЗ

При вирішенні задачі збереження самого факту листування автоматизація роботи алгоритму дозволяє користувачеві значно заощадити час. Розробка спеціального додатка створює умови для приховування листування інформацією з обмеженим доступом шляхом надсилання фотографій.

Можна надіслати приховану інформацію, наприклад, вітальну листівку. Нікому неможливо візуально визначити факт прихованої інформації.

Запропонований в кваліфікаційній роботі алгоритм роботи модуля можна описати так:

1. перевірка наявності графічного зображення в листі;
2. перевірка вкладення на наявність якогось обумовленого стеганоалгоритмічного підпису;
3. повідомити користувача про наявність пакету;
4. отримання повідомлення з конфіденційною інформацією.

Як і надбудова Microsoft Outlook, надбудови браузера можуть використовувати стеганографію для:

- завантаження файлів зображень на веб-ресурс;
- завантаження файлів зображень із веб-ресурсу;
- використання веб-інтерфейсу для надсилання файлів зображень.

Для вирішення цих проблем рекомендується використовувати розроблену програму, вбудовану в браузер.

Коли виявляється нова фотографія, користувач завантажує її на свій комп’ютер за допомогою веб-браузера чи іншого програмного забезпечення. Потім алгоритм отримує приховану інформацію із зображення.

Ця процедура дозволяє створити секретний механізм обміну повідомленнями з цифровими фотографіями. Сторонні спостерігачі сприймуть це як простий обмін фотографіями.

Рівень інтерфейсу — це рівень відображення даних на стороні клієнта. Дані подаються користувачеві за допомогою графічного інтерфейсу у вигляді Windows Forms.

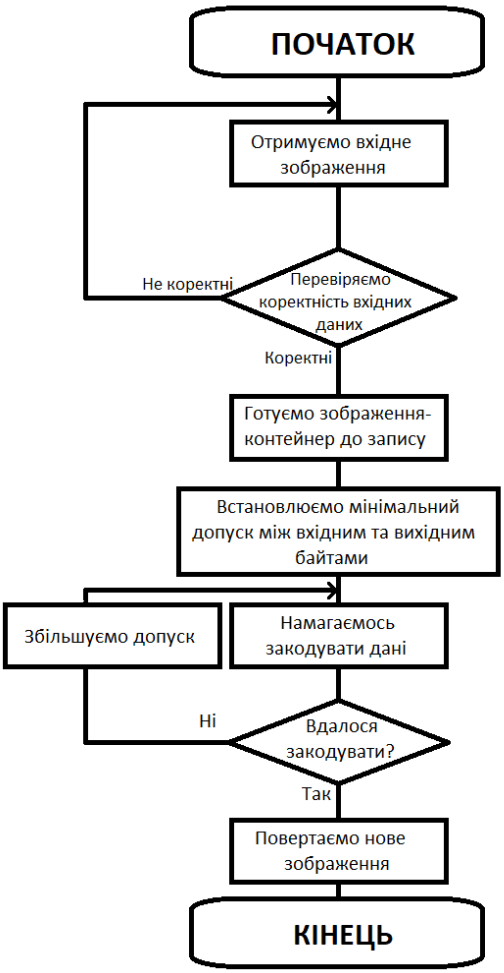


Рисунок 4.2 – Алгоритм роботи додатку

Дана програма буде виконана на основі власного стеганографічного алгоритму та працюватиме з файлами зображень у форматі jpg.

У даному підході інформація буде приховуватись у пікселях зображення таким чином, щоб елемент з зашифрованим байтом інформації мінімально відрізнявся від початкового.

Робота з пікселями буде проводитись у кольоровій палітрі RGB, де один з каналів (довільний, у нашому випадку оберемо канал Blue) – технічний, він інформуватиме чи захований у даному пікселі інформація, чи ні. Інших два канала – несуть інформацію: якщо якийсь з них буде в діапазоні $R - t < X < R + t$, або $G - t < X < G + t$, де R або G – це, відповідно, один з каналів, X – це наш байт прихованих даних, а t – це допуск, який ми вказуємо. Таким чином інформація буде записуватися лише в ті байти, де вона викликатиме найменші спотворення.

Перед записом інформації у файл зображення, його необхідно підготувати, а саме – сформувати технічний канал. Цей канал повинен інформувати про те, чи є в даному пікселі захована інформація, і якщо така інформація є, то у якому каналі.

Отже, наш технічний синій канал повинен мати три можливих стани. Реалізуємо це за допомогою ділення по модулю 3:

- якщо B % 3 = 0 – значить у даному пікселі нічого немає,
- якщо B % 3 = 1 – значить байт заховано у червоному каналі,
- якщо B % 3 = 2 – значить байт заховано у зеленому каналі.

Відповідно, підготовка контейнера заключатиметься в тому, щоб зробити синій канал кожного пікселя кратним 3. Кожен байт зміниться максимум на 2 одиниці, по синьому каналу, а така незначна зміна, на практиці, ніяк візуально не змінить зображення.

Орієнтуючись на це, перейдемо до програмної реалізації алгоритму.

Реалізація буде виконана на мові C#, у об’єктно-орієнтованій парадигмі.

```
class Stegano
{
    Bitmap photo;      // Інкапсульовані змінні: зображення та дані
    byte[] openData;

    // GET-метод для отримання зображення як масиву байт
    public byte[] GetPhotoBytes()
    {
        using (var ms = new MemoryStream())
        {
            photo.Save(ms, photo.RawFormat);
            return ms.ToArray();
        }
    }

    // GET-метод для отримання повідомлення
    public byte[] GetOpenData()
    {
        if (openData == null)
        {
            throw new Exception("Дані не задано");
        }
        else
        {
            return openData;
        }
    }
```

Відповідно потрібно організувати конструктори для кодування та декодування інформації.

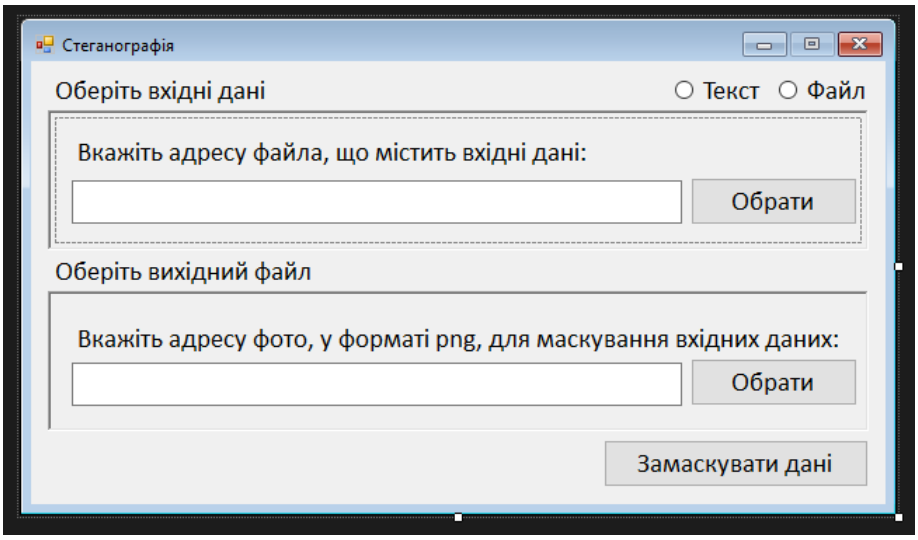


Рисунок 4.3 – Екрана форма

Детальна структура програмного забезпечення наведена у додатку А.

4.2 Розробка програмного додатка на основі запропонованого алгоритму

Для зображень BMP реалізовано метод заміни найменшого значущого біта. Цей метод критично нестійкий до будь-якої деформації контейнера, а також до його стиснення, що знищує всю приховану інформацію. Крім того, введення інформації таким чином легко виявити за допомогою стеганографічних атак. Але перекодування в JPEG долає ці труднощі.

Щоб краще захистити інформацію, приховану в зображенні шляхом заміни, доцільно використовувати криптографічні методи, які шифрують повідомлення до того, як воно буде приховано, і, таким чином, інформація буде захищена різними рівнями безпеки, які приховають факт передачі повідомлення, та ускладнюються її шифруванням

Крім криптографії, ви можете використовувати алгоритми для вибору пікселя та його розташування для вбудовування біта повідомлення за певним правилом. Такий підхід, порівняно з послідовним вибором пікселів для приховування інформації, захистить повідомлення від розшифровки, навіть якщо в контейнері будуть виявлені зайві дані, оскільки біти повідомлення будуть випадково видалені з зображення за допомогою алгоритму.

Рисунок 4.3 – Передача прихованих даних

Ця процедура дозволяє організувати секретний механізм обміну повідомленнями з цифровими фотографіями. Сторонні спостерігачі сприймуть це як звичайний обмін фотографіями.

Розміром можна керувати, ретельно вибираючи контейнери. Підготуйте необхідні контейнери, відзначивши, в які розміри конкретні кодуючі модулі перетворять фотографії. На жаль, методи форматування зазвичай не вирішують проблеми стійкості до стиснення вбудованої інформації. Це пояснюється тим, що, використовуючи сегментні маркери під час реалізації інформації, можна очікувати, що завантажувач без потреби знищить ці маркери навіть до дискретного косинусного перетворення [21].

Адже алгоритм впроваджує конфіденційне повідомлення в сегменти, що пропускаються утилітами читання JPEG-зображень, приміром DAC; COM; APP15, SOF2 - SOF10 чи інші [18].

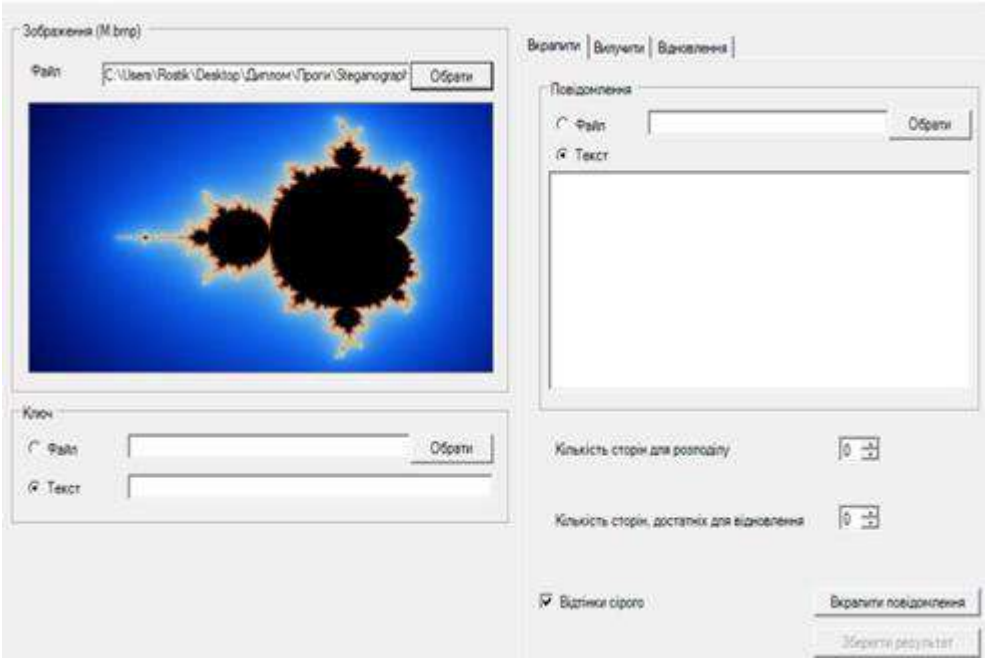


Рисунок 4.4 – Перетворення файлу формату BMP

Наступним кроком є визначення тексту повідомлення. Це виконується одним із двох можливих способів. Перший метод передбачає, що користувач вводить текст повідомлення з клавіатури, тоді як другий метод вимагає, щоб повідомлення було у текстовому файлі .txt. Маленькі повідомлення легше вказати самостійно, тоді як великі легко завантажити з файлу. Також необхідно

звернути увагу на розмір контейнера, щоб приблизно зрозуміти максимально можливу довжину прихованого повідомлення, яке буде оброблено алгоритмом і заховане в контейнер.

Перед вставленням повідомлення стискається, далі йде контейнер JPEG:

1. виконується аналіз поточної файлової структури. При цьому виділяються межі сегментів;

2. створення назви нового сегмента, наприклад COM чи APP15;

3. створений фрагмент заповнюється байтами зі стиснутого повідомлення;

4. до структури файлу JPEG додано новий елемент.

Потім зображення з прихованої конфіденційною інформацією надсилається на потрібну адресу.

Скріншот форми для отримання прихованого конфіденційного повідомлення наведено на рис. 4.5.

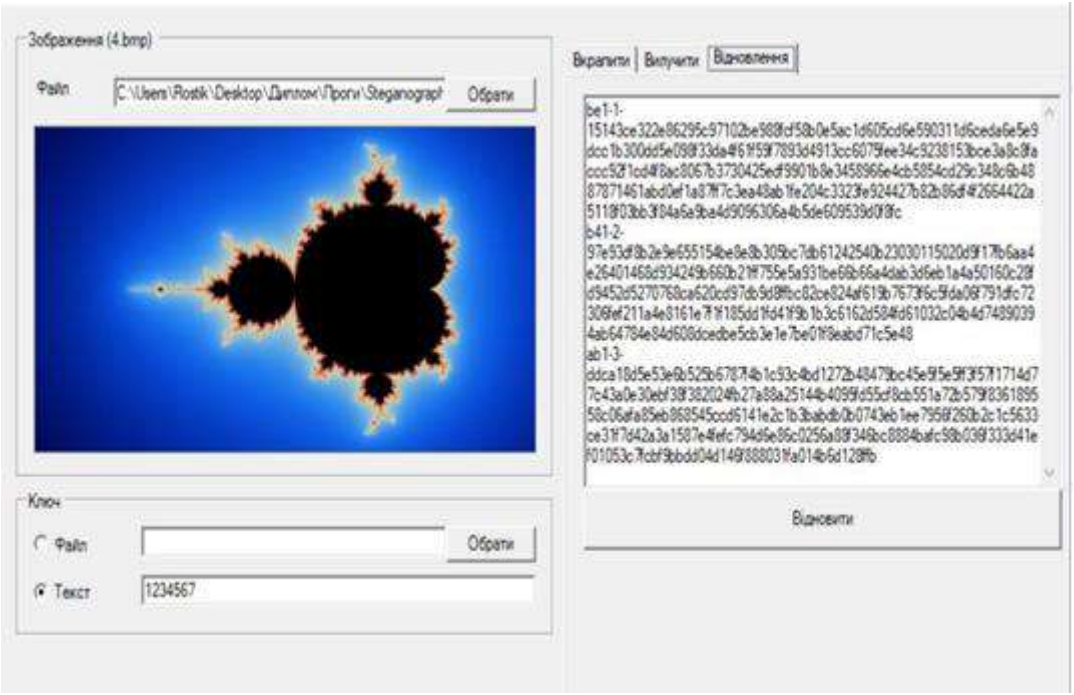


Рисунок 4.6 – Заповнення форми для екстракції повідомлення

Під час тестування суттєвих недоліків у програмному застосуванні комп’ютерної стеганографії для контейнерів у вигляді зображень не виявлено,

що свідчить про високу якість розробки та доцільність впровадження при необхідності.

4.3 Порівняльний аналіз стеганоалгоритмів

Розроблений алгоритм порівнювався з існуючими JPHS та JSteg реалізованими в програмному додатку S-tools [18]. Вони порівнювалися по максимальному розміру даних (txt-файл) які можна приховати в JPEG- файл об'ємом 2550212 байт.

Для якісного порівняння роботи програм, результати їх роботи будемо заносити в таблиці, описуючи такі параметри: формат файлу, швидкість виконання, оригінальне зображення, зображення після кодування, початковий та кінцевий розмір файлу.

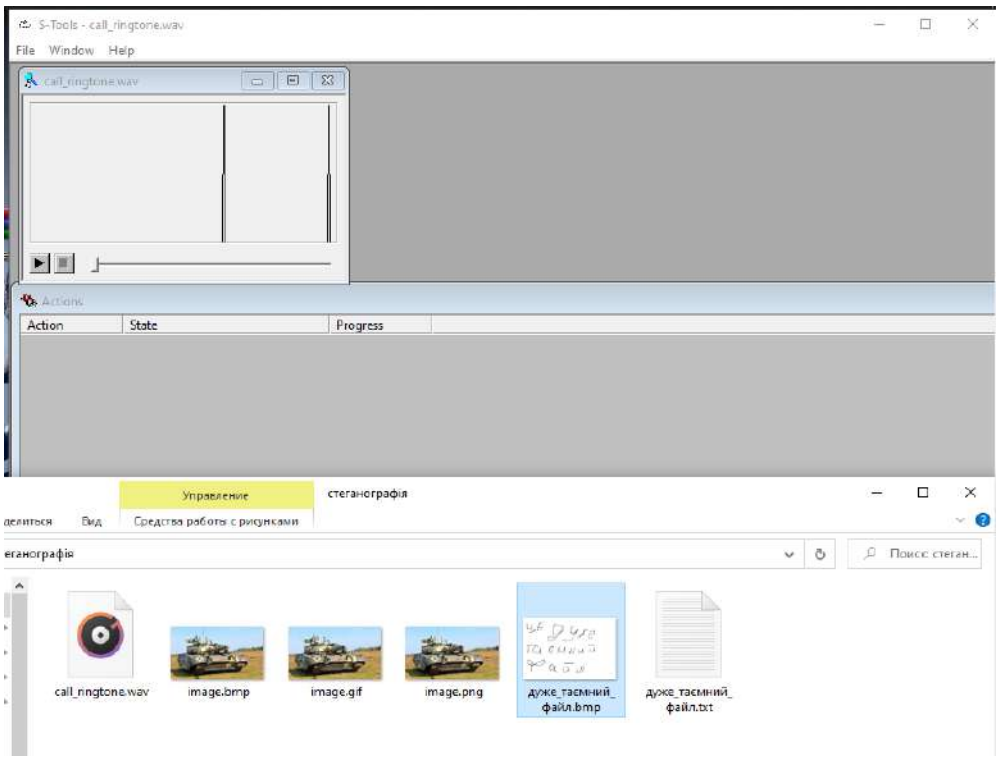


Рисунок 4.7 – Робота програми S-tools

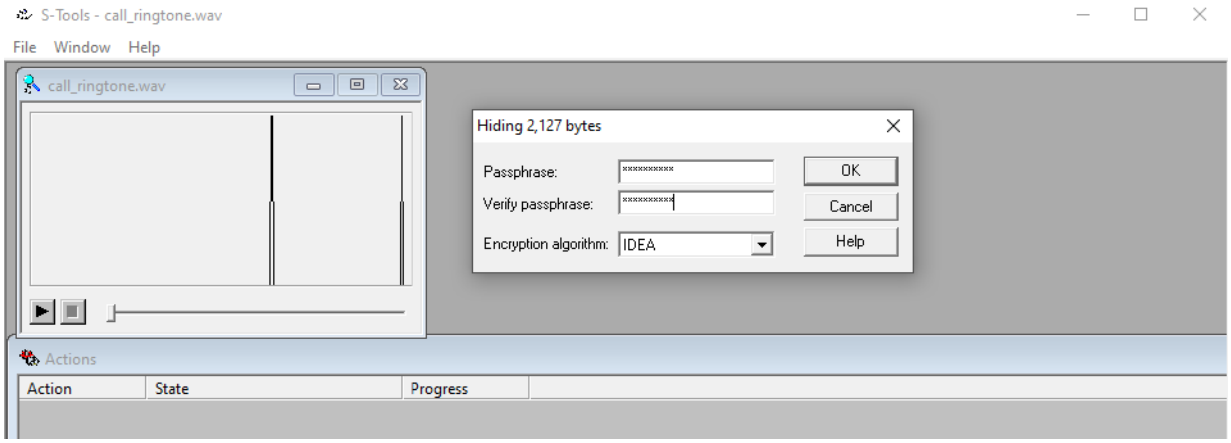


Рисунок 4.8 – Вибір алгоритмів у програмі S-tools

Оскільки файл даних більший ніж контейнер, але, не дивлячись на це, програма успішно кодує та розкодовує дані, можна припустити, що файл даних попередньо архівується.

Отже, дана програма успішно приховує дані в контейнерах, не змінюючи при цьому їх розмір. Контейнер дещо змінюється, що помітно на скріншотах у таблиці, проте це не є особливо помітним для звичайного користувача, у якого немає початкового файла.

Результати порівняння алгоритмів представлено в таблиці 4.1 :

Таблиця 4.1 - Результати порівняння алгоритмів

№	Алгоритм	Розмір повідомлення
1	Модифікований алгоритм (4 біти)	3538943
2	Модифікований алгоритм (3 біти)	2654208
3	Модифікований алгоритм (2 біти)	1769472
4	Модифікований алгоритм (1біт)	884736
5	Jsteg	347747
6	JPHS	38502

Зобразимо одержані результати у вигляді гістограми (рис. 4.9).

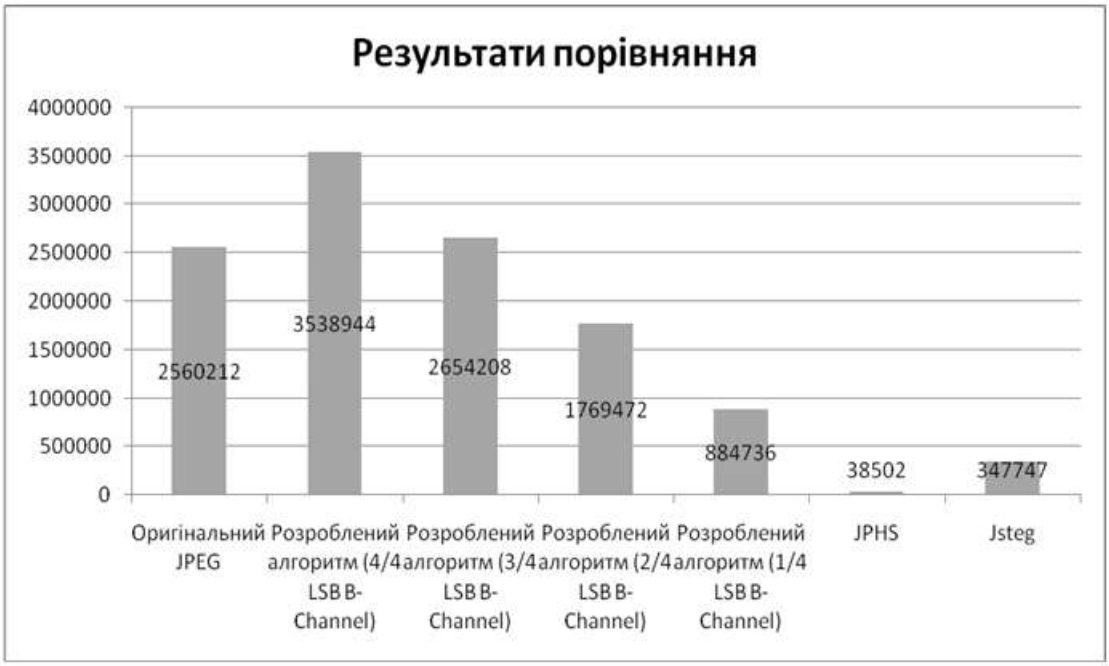


Рисунок 4.9 - Графік порівняння розробленого і існуючих стеганографічних алгоритмів

Крім того, з графіку видно, що:

1. При використані режиму 4/4 LSB, розроблений стеганоалгоритм впроваджує інформацію з обмеженим доступом, об'єм якої більше стеганоконтейнера формату JPEG.

2. При використанні режими 2/4, 3/4 і 4/4 LSB, запропонований в роботі стеганоалгоритм впроваджує інформацію з обмеженим доступом, об'єм якої можна вимірювати в мегабайтах.

4.4 Висновки

У цьому розділі наведено сери застосунку розробленого стеганоалгоритму. Це такі завдання, як забезпечення конфіденційності обміну повідомленнями та захист авторських прав. Для зображень BMP реалізовано метод заміни найменшого значущого біта. Цей метод критично нестійкий до будь-якої

деформації контейнера, а також до його стиснення, що знищує всю приховану інформацію. Крім того, введення інформації таким чином легко виявити за допомогою стеганографічних атак. Але перекодування в JPEG долає ці труднощі.

Крім того, під час виконання кваліфікаційної роботи зроблено висновки що запропонована система секретної передачі інформації в соцмережі є альтернативним рішенням із криптографічними засобами захисту каналів зв'язку (використання зашифрованого протоколу SSL/TLS). Реалізація веб-браузерів і поштових клієнтів тісно пов'язана з системами прийняття рішень.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розв’язано актуальну практичну задачу в галузі кібербезпеки, і отримано певний прикладний результат у вигляді функціонально-придатного програмного застосунку.

Запропонований алгоритм використовує подвійне перетворення формату з використанням сервісних сегментів формату JPEG. Інформація з обмеженим доступом маскується в модифікованому потоці байтів BMP, а службові дані, необхідні для вилучення, записуються як частина звичайної структури сегмента. Цей підхід зробив алгоритм актуальним.

Основні результати кваліфікаційної роботи:

1. Досліджено види стеганографічних алгоритмів та тенденції та закономірності розвитку стеганосистем.

2. Проведено аналіз можливості застосування різних підходів до розробки стеганоалгоритмів.

3. Запропоновано алгоритм, який дає змогу вбудовувати великі об’єми інформації з обмеженим доступом в цифрове зображення заповненого контейнера на стороні передачі та отримання цієї інформації на стороні прийому.

4. Запропонований алгоритм, дозволяє використовувати просторову зону файлу зображення та враховано можливість його функціонування в умовах втрати бітів під час міжформатних перетворень та стиснення.

5. Розроблено методику оцінки спотворень зображення.

6 На основі використання стеганографічного модуля розроблено програмне забезпечення.

7. Виконано порівняльний аналіз за критерієм розміру можливого приховування конфіденційної інформації популярних алгоритмів JPHS та JSteg та запропонованого.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Комп’ютерна стеганографічна обробка та аналіз мультимедійних даних: підручник. / Г. Ф. Конахович, Д. О. Прогонов, О. Ю. Пузиренко. – Київ: «Центр учбової літератури», 2018. – 558 с

2. Баранник В. В. Основы теории структурно-комбинаторного стеганографического кодирования: монография / В. В. Баранник, А. Э. Бекиров, Д. В. Баранник. – Х. : ХНУРЕ, 2017. - 256 с

3. Кузнецов О. О. Стеганографія : навчальний посібник / О. О. Кузнецов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2011. – 232 с.

4. НД ТЗІ 3.6-001-2000 Технічний захист інформації. Комп’ютерні системи. Порядок створення, впровадження, супроводження та модернізації засобів технічного захисту інформації від несанкціонованого доступу

5. Handbook of the Image-Based Security Techniques / Editors: Shivendra Shivani, Suneeta Agarwal, Jasjit S. Suri. – Taylor & Francis Group, 2018. – 443 p

6. Інформаційна безпека: навчальний посібник / [Ю. Я. Бобало, І. В. Горбатий, М. Д. Кіселичник, А. П. Бондарєв та ін.]; за заг. ред. д-ра техн. наук, проф. Ю. Я. Бобала та д-ра техн. наук, доц. І. В. Горбатого. – Львів: Видавництво Львівської політехніки, 2019. – 580 с.

7. Лейман А.В. Аналіз принципів створення і роботи стеганографічних алгоритмів / Лейман А.В., Коробейников А.Г., Кувшинов С.С., Блинов С.Ю.//Програмні системи і обчислювальні методи. - Москва: М: "НБ-Медиа", 2012. - Вип. l. - № 1. - Моделі і методи управління інформаційною безпекою. - С. 28 - 36. - 102 с. - ISSN 2305-6061.

8. Коробейников А.Г. Вплив зовнішніх дій на DC коефіцієнти матриці ДВП в півтонових зображеннях / Коробейников А.Г., Прохожев Н.Н., Михайличенко О.В. // Науково-технічний вісник СПб ГУ ІТМО. Інформаційні технології і телекомунікаційні системи. Т. 56, 2008. - стр. 57-62.

9. Mathematical Structures For Computer Graphics / Edited by the Steven J. Janke. – John Wiley & Sons, Inc, 2015. – 410 p

10. Steganography Techniques for Digital Images / Edited by the Abid Yahya.

– Springer International Publishing AG, 2019. – 131 p

11. Digital Watermarking and Steganography: Fundamentals and Techniques / Edited by Frank Y. Shih. – Taylor & Francis Group, 2017. – 293 p

12. НД ТЗІ 1.1-005-07 Захист інформації на об’єктах інформаційної діяльності. Створення комплексу технічного захисту інформації. Основні положення

13. Конахович Г.Ф. Комп'ютерна стеганографія. Теорія та практика / Конахович Г.Ф., Пузиренко А.Ю. - Київ: МК-Пресс, 2006. - 288 с.

14. Pro Processing for Images and Computer Vision with OpenCV Solutions for Media Artists and Creative Coders / Edited by the Bryan WC Chung. – Academy of Visual Arts, Kowloon Tong, Hong Kong, 2017. – 301 p

15. Secure Digital Documents Using Steganography and QR Code dissertation / Edited by Mohamed Samah Hassanein. – Department of Computer Science, Brunel University, 2014. – 191 p

16. Multimedia Security: Watermarking, Steganography, and Forensics / Edited by Frank Y.Shih. – Taylor & Francis Group, 2013. – 411 p.

17. Efficient And Robust Video Steganography Algorithms for Secure Data Communication; dissertation / Edited by Ramadhan J. Mstafa. – The school of engineering University of Bridgeport Connecticut, 2017. – 165 p

18. HTML Steganography Algorithms and the Detection Methods dissertation / Edited by Iman Thannoon Sedeeq. – the University of Liverpool, 2018. – 140 p.

19. Гребенников В. Стеганография. История тайнописи / В. Гребенников – Москва: ЛитРес: Самиздат, 2019. – 142 с

20. Image Steganography Based on the Discrete Wavelet Transform and Enhancing Resilient Backpropogation Neural Network dissertation / Edited by Ahmed Shihab Ahmed AL- Naima. – Middle East University Amman- Jordan, 2015. – 113 p

21. Основи двовимірної комп’ютерної графіки : навчальний посібник / О. О. Сафранова, К. В. Донець. – К. : КНУТД, 2016. – 175 с.

22. Муляр І.В. Ітераційно-геометричний метод для стійкого перцептуального хешування зображення / В. М. Джулій, Ю. П. Кльоц, І. В.

Муляр, В. М. Чешун // Вісник Хмельницького національного університету. Технічні науки. – 2020. – № 1. – С. 76–79

23. Муляр І.В. Симетрична криптосистема із нелінійним шифруванням та можливістю створення шифротексту з метою маскування / В. А. Анікін, В. М. Джулій, І.В. Муляр, В.С. Орленко, В.Ю. Тітова // Вісник Хмельницького національного університету. Технічні науки. – 2020. – № 6. – С. 12-16

24. НД ТЗІ 3.3-001-07 Захист інформації на об’єктах інформаційної діяльності. Створення комплексу технічного захисту інформації. Порядок розроблення та впровадження заходів із захисту інформації

ДОДАТОК А

(обов’язковий)

Фрагмент коду програмної реалізації стеганографічного алгоритму

```
using System;
using System.Drawing;
using System.IO;
using System.Text;
using System.Windows.Forms;
```

Фрагмент коду програмної реалізації стеганографічного алгоритму……

```
namespace Стеганографія
{
    public partial class Form1 : Form
    {
        bool isText = true;

        public Form1()
        {
            InitializeComponent();
        }

        private void RadioButton1_CheckedChanged(object sender, EventArgs e)
        {
            panel2.Visible = true;
            panel3.Visible = false;
            isText = true;
        }

        private void RadioButton2_CheckedChanged(object sender, EventArgs e)
        {
            panel3.Visible = true;
            panel2.Visible = false;
            isText = false;
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            string path;

            OpenFileDialog fDialog = new OpenFileDialog();
            fDialog.Title = "Оберіть файл";
            fDialog.Filter = "Всі формати|*.*";
            fDialog.InitialDirectory = @"D:\";
            if (fDialog.ShowDialog() == DialogResult.OK)
            {
                path = fDialog.FileName.ToString();
            }
            else
                return;
            textBox2.Text = path;
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            this.MaximumSize = new Size(1500, 355);
            this.MinimumSize = new Size(625, 355);
```

```
}

private void Button2_Click(object sender, EventArgs e)
{
    string path;

    OpenFileDialog fDialog = new OpenFileDialog();
    fDialog.Title = "Оберіть файл";
    fDialog.Filter = "PNG зображення|*.png";
    fDialog.InitialDirectory = @"D:\";
    if (fDialog.ShowDialog() == DialogResult.OK)
    {
        path = fDialog.FileName.ToString();
    }
    else
        return;
    textBox3.Text = path;
}

private void Button3_Click(object sender, EventArgs e)
{
    try
    {
        byte[] data;
        if (isText)
        {
            data = Encoding.UTF8.GetBytes(textBox1.Text);
            Array.Resize(ref data, data.Length + 1);
            data[data.Length - 1] = 125;
        }
        else
        {
            using (FileStream fstream = File.OpenRead(textBox2.Text))
            {
                data = new byte[fstream.Length];
                fstream.Read(data, 0, data.Length);
            }
            string fName = Path.GetFileName(textBox2.Text);
            byte[] fN = Encoding.UTF8.GetBytes(fName);
            byte[] ln = BitConverter.GetBytes(fN.Length);

            Array.Resize(ref data, data.Length + fN.Length);
            Array.Copy(fN, 0, data, data.Length - fN.Length, fN.Length);
            Array.Resize(ref data, data.Length + 4);
            Array.Copy(ln, 0, data, data.Length - 4, 4);

            Array.Resize(ref data, data.Length + 1);
            data[data.Length - 1] = 126;
        }

        byte[] arch_data = Module1723.ArchivSimple_8bit(data, true);

        if (arch_data.Length < data.Length)
        {
            data = arch_data;
            Array.Resize(ref data, data.Length + 1);
            data[data.Length - 1] = 141;
        }
        else
        {
            Array.Resize(ref data, data.Length + 1);
```

```
    data[data.Length - 1] = 123;
}

Stegano stegano;

using(Image img = Image.FromFile(textBox3.Text))
{
    stegano = new Stegano(data, Image.FromFile(textBox3.Text));
}


bool corr = false;
try
{
    corr = stegano.EncodeData(5);
}
catch (Exception)
{
    try
    {
        corr = stegano.EncodeData(10);
    }
    catch (Exception)
    {
        try
        {
            corr = stegano.EncodeData(15);
        }
        catch (Exception)
        {
            try
            {
                corr = stegano.EncodeData(25);
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Помилка");
                return;
            }
        }
    }
}
if (corr)
{
    string kyda;
    FolderBrowserDialog fDialog1 = new FolderBrowserDialog();
    if (fDialog1.ShowDialog() == DialogResult.OK)
    {
        kyda = fDialog1.SelectedPath.ToString();
    }
    else
        return;

    kyda += "\\" + NameGenerate(kyda);

    data = stegano.GetPhotoBytes();

    using (FileStream fstream = new FileStream(kyda, FileMode.Create))
    {
        fstream.Write(data, 0, data.Length);
    }
```

```
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Помилка");
            }

        }
        private string NameGenerate(string path)
        {

            string name = "crypt.png";

            if (File.Exists(path + "\\" + name))
            {
                int i = 1;
                while (System.IO.File.Exists(path + "\\" + name))
                {
                    name = "crypt(" + i + ").png";
                    i++;
                }
                return name;
            }
            else
            {
                return name;
            }
        }
    }
}


          class Stegano
          {
              Bitmap photo;      // Інкапсульовані змінні: зображення та дані
              byte[] openData;

              // GET-метод для отримання зображення як масиву байт
              public byte[] GetPhotoBytes()
              {
                  using (var ms = new MemoryStream())
                  {
                      photo.Save(ms, photo.RawFormat);
                      return ms.ToArray();
                  }
              }

              // GET-метод для отримання повідомлення
              public byte[] GetOpenData()
              {
                  if (openData == null)
                  {
                      throw new Exception("Дані не задано");
                  }
                  else
                  {
                      return openData;
                  }
              }
```

```
        // Конструктор класу для закодування
        public Stegano(byte[] openData, Image photo)
        {
            this.photo = (Bitmap)photo;
            this.openData = openData;
        }

        // Конструктор класу для розкодування
        public Stegano(Image photo)
        {
            this.photo = (Bitmap)photo;
            this.openData = null;
        }
 }


public bool EncodeData (int tolerance)
   {
        // Перевіряємо на коректність дані
      Exception nullPhoto = new Exception("Розмір цільового фото не достатній");

      if (this.photo == null)
          throw new Exception("Цільове фото не задано");
      if (this.openData == null || this.openData.Length == 0)
          throw new Exception("Відсутня вхідна інформація");
      if ((this.photo.Width * this.photo.Height) / 2 < this.openData.Length)
          throw nullPhoto;



        // Проводимо підготовку контейнера до запису
      Bitmap clearPhoto = ClearPhoto1((Bitmap)this.photo);

      int currentPoint = 0;

      byte[,] pixels = new byte[clearPhoto.Width * clearPhoto.Height, 2];
      int c = 0;


        // Запускаємо цикл по пікселях зображення
      for (int i = 0; i < clearPhoto.Height; i++)
      {
          for (int j = 0; j < clearPhoto.Width; j++, c++)
          {
              pixels[c, 0] = clearPhoto.GetPixel(j, i).R;
              pixels[c, 1] = clearPhoto.GetPixel(j, i).G;

              int t1_ = (pixels[c, 0] - this.openData[currentPoint]);
              if (t1_ < 0)
                  t1_ = -t1_;

              int t2_ = (pixels[c, 1] - this.openData[currentPoint]);
              if (t2_ < 0)
                  t2_ = -t2_;

              if ((t1_ < tolerance) || (t2_ < tolerance))
              {
                  byte r_ = (byte)t1;
                  byte g_ = (byte)t2;
                  int b_ = clearPhoto.GetPixel(j, i).B;
                  if (r_ < g_) // В червоний
                  {
```

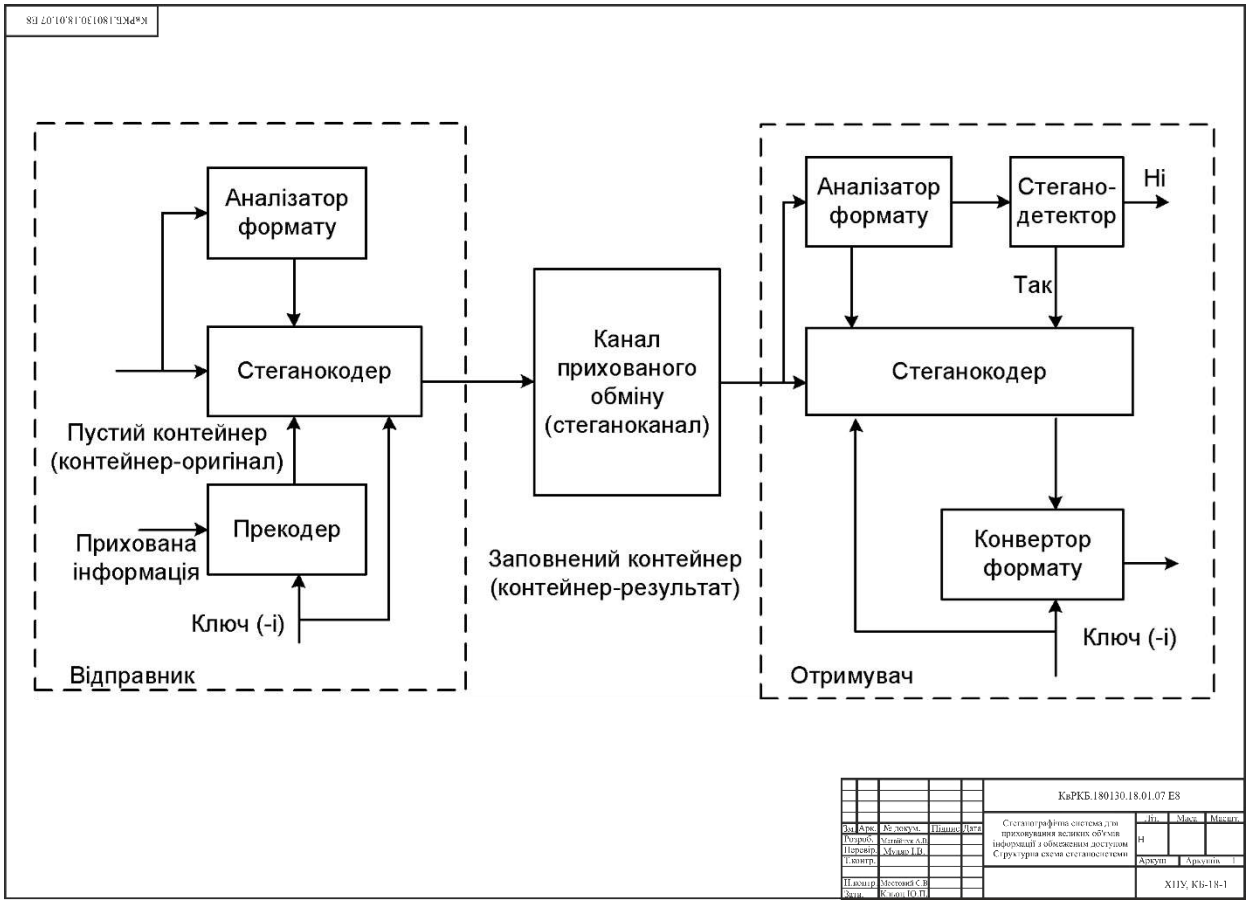
```
                        b_++;
                        if (b_ > 255)
                            b_ -= 3;

                        clearPhoto.SetPixel(j, i, Color.FromArgb(
                            this.openData[currentPoint],
                            pixels[c, 1],
                            b_
                            ));
                    }
                    else // або в зелений
                    {
                        b_ += 2;
                        if (b_ > 255)
                            b_ -= 3;
                        clearPhoto.SetPixel(j, i, Color.FromArgb(
                            pixels[c, 0],
                            this.openData[currentPoint],
                            b_
                            ));
                    }
                    currentPoint++;
                    if (currentPoint == this.openData.Length)
                    {// Якщо всі необхідні байти вже зашифровано – закінчуємо роботу
                        this.photo = clearPhoto;
                        return true;
                    }

                }
            }
        }
        throw nullPhoto;
    }
public void DecodeData()
        {
// Перевіряємо на коректність дані
            Exception nullPhoto = new Exception("Розмір цільового фото не достатній");

            if (this.photo == null)
                throw new Exception("Цільове фото не задано");

            byte[] data = new byte[0];
            int c = 0;

            Bitmap photo = new Bitmap(this.photo);
            Color[] pixels = new Color[photo.Height * photo.Width];
// Запускаємо цикл по пікселях зображення
            for (int i = 0; i < this.photo.Height; i++)
            {
                for (int j = 0; j < this.photo.Width; j++, c++)
                {
                    pixels[c] = photo.GetPixel(j, i);

                    if ((pixels[c].B % 3) != 0)
                    {
                        Array.Resize(ref data, data.Length + 1);
                        if (pixels[c].B % 3 == 1)
                        {
                            data[data.Length - 1] = photo.GetPixel(j, i).R;
                        }
                        else
```

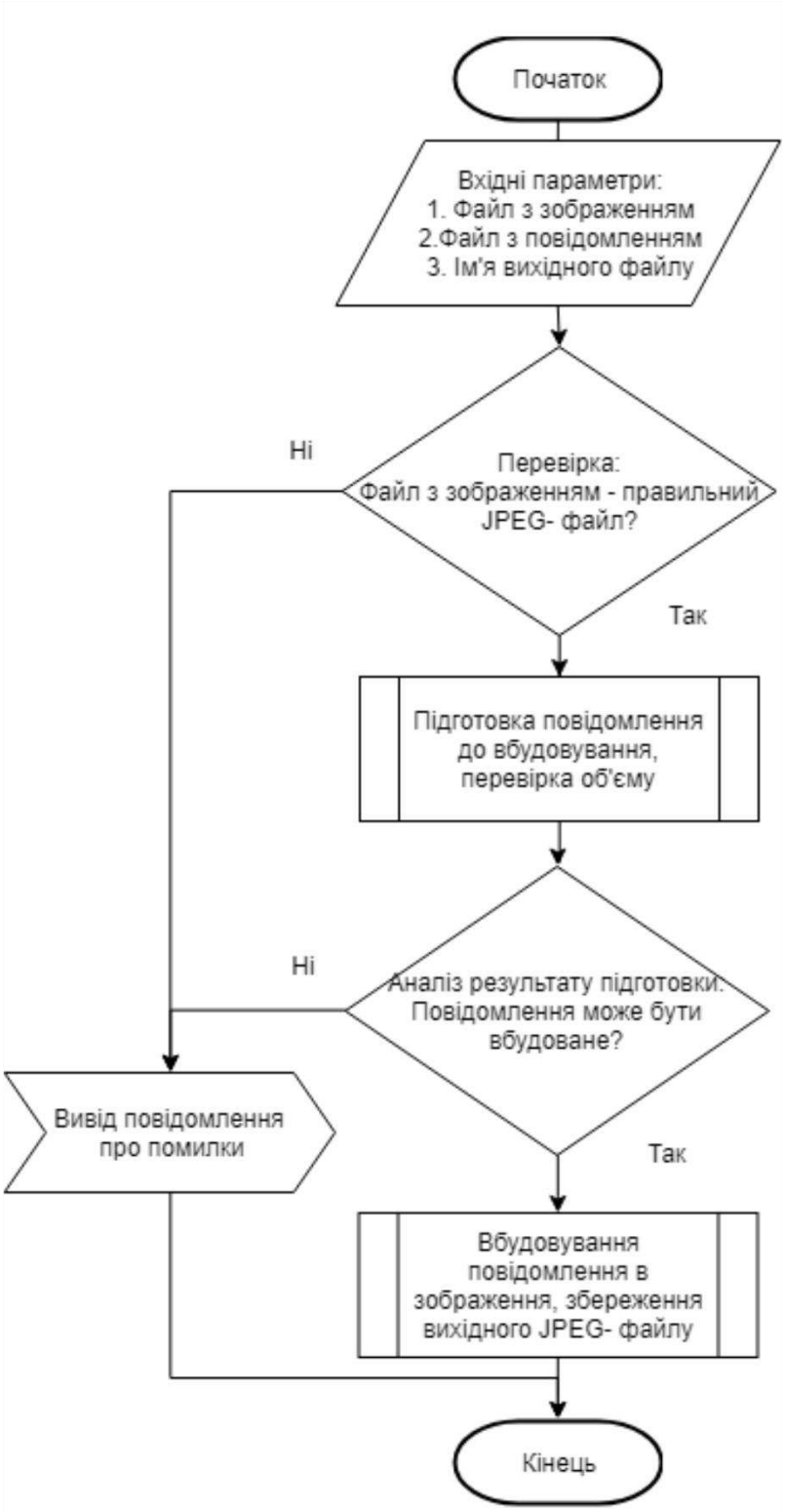
```
                {
                    data[data.Length - 1] = photo.GetPixel(j, i).G;
                }
            }
        }
    }
    this.openData = data;
    return;
}
```

ДОДАТОК Б

(обов’язковий)

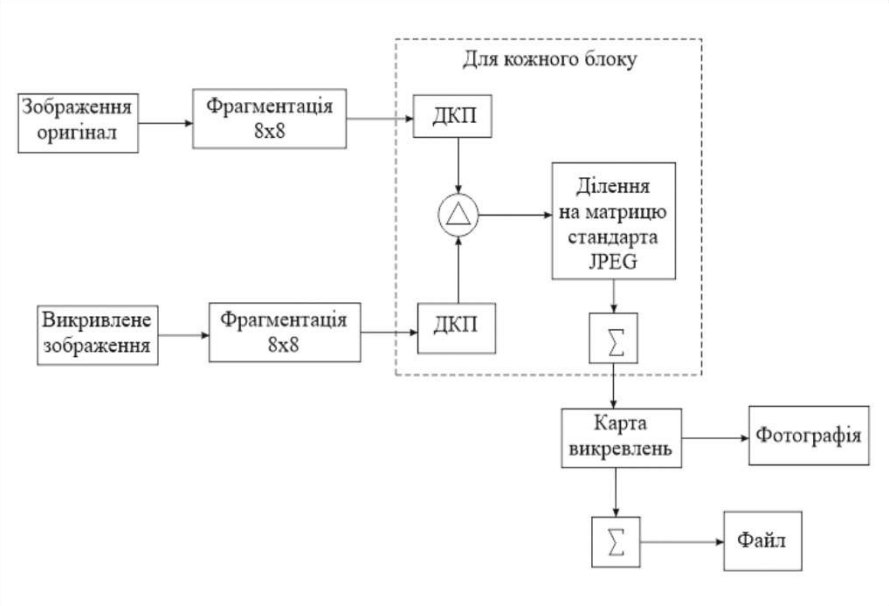
Копія графічної частини





					КвРКБ.180130.18.01.07 Е8		
Зм.	Арк.	№ докум.	Підпис	Дата	Стеганографічна система для приховування великих об'ємів інформації з обмеженим доступом Стеганоалгоритм	Літ.	Маса / Масшт.
Розроб.		Матвійчук А.В.				Н	
Перевір.		Мудрак І.В.				Аркуш	Аркушів 1
Т.контр.							
Н.контр.		Мостовий С.В.				ХНУ, КБ-18-1	
Затв.		Кльоц Ю.П.					

Блок-схема алгоритму оцінки змін фотографії із застосуванням матриці стандарту JPEG



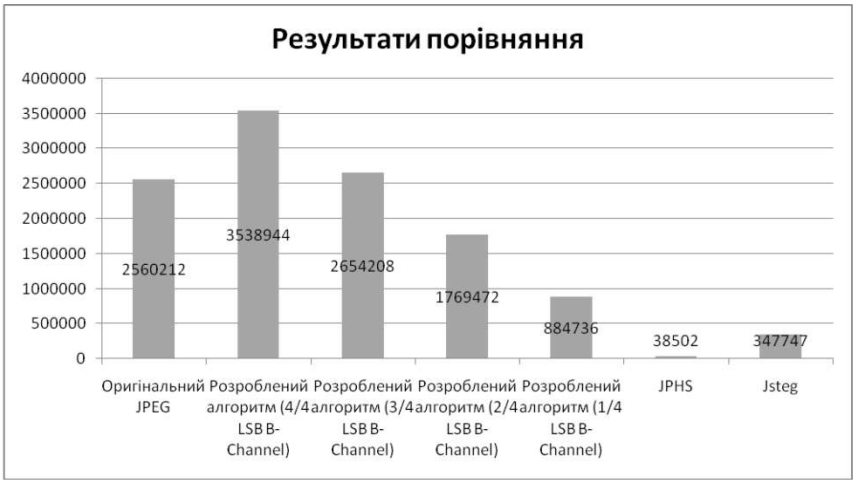
Коефіцієнти квантування матриці стандарту JPEG

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Залежність розміру впроваджуваної інформації

№	Байтові послідовності	Файл 1	Файл 2	Файл 3
1	Зображення JPEG	297641	2561232	4339765
2	Проміжний BMP	2102702	21244718	40254950
3	Кількість біт синього каналу -1	87572	885735	1682604
4	Кількість біт синього каналу -2	175274	1779462	3355418
5	Кількість біт синього каналу -3	262757	2664209	5028122
6	Кількість біт синього каналу -4	350208	3548844	6691716

Результати порівняння з алгоритмами JPHS і JSteg



					КвРКБ.180130.18.01.07 ЕЗ		
Зм.	Арк.	№ докум.	Підпис	Дата	Стеганографічна система для приховування великих об'ємів інформації з обмеженим доступом Результати тстування додатку	Літ.	Маса / Масшт.
Розроб.		Матвійчук А.В.				Н	
Перевір.		Мудяр І.В.				Аркуш	Аркушів 1
Т.контр.							
Н.контр.		Мостовий С.В.				ХНУ, КБ-18-1	
Затв.		Кльоц Ю.П.					

Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльоцу Ю.П.
Матвійчук Андрій Вікторович
ПІБ здобувача вищої освіти

студента ФІТ, 4 курсу, групи КБ-18-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

6.06.2022
дата

підпис

UNICHECK

Ім'я користувача:
Кафедра кібербезпеки

ID перевірки:
1011462123

Дата перевірки:
04.06.2022 19:13:21 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
04.06.2022 19:16:14 EEST

ID користувача:
100008300

Назва документа: **матвійчук_плагіат**

Кількість сторінок: **68** Кількість слів: **10835** Кількість символів: **83762** Розмір файлу: **2.33 MB** ID файлу: **1011340401**

12.9%
Схожість

Найбільша схожість: **8.32%** з джерелом з Бібліотеки (ID файлу: **1008437104**)

11.8% Джерела з Інтернету	56	Сторінка 70
11.6% Джерела з Бібліотеки	117	Сторінка 70

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	14

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 0.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 9%**

	Документ		Суммарное совпадение по Базе Данных	
ID: 104454 Название: Стеганографічна система для приховування великих об’ємів інформації з обмеженим доступом Добавлено в БД: 2022-06-04 Авторы: Матвійчук Андрій Вікторович Руководители: Муляр Ігор Консультанты: Опоненты:	Символы	Лексемы	Символы	Лексемы
	62316	958	1153 (2%)	17 (2%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Стеганографічна система для приховування великих об'ємів інформації з обмеженим доступом

Автор: Матвійчук Андрій Вікторович

Спеціальність: 125 – Кібербезпека

Освітня програма: Кібербезпека

Керівник: Муляр Ігор Володимирович, к.т.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 12.9 % що, з урахуванням наведених обгрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ І.В. Муляр

Гарант освітньої програми _____ В.М. Чешун

Завідувач кафедри КбКСМ, гарант ОП _____ Ю.П. Кльоц

Дата: 06.06.2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «бакалавр»

Студент Матвійчук Андрій Вікторович

Тема Стеганографічна система для приховування великих об'ємів інформації з обмеженим доступом

Спеціальність 125 – Кібербезпека

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «бакалавр»:

кількість листів креслень 4; кількість сторінок записки 67.

1. Короткий зміст роботи та прийнятих рішень У кваліфікаційній роботі розроблено алгоритм приховування великого об'єму даних в просторовій області зображення формату JPEG за рахунок компенсації втрачених бітів під час міжформатного перетворення, що аналізує і змінює структуру сегментів файлів. Розроблений застосунок створює умови для приховування кореспонденції шляхом надсилання фотографій в поштових клієнтах, соціальних мережах.

2. Висновок про відповідність кваліфікаційної роботи завданню Кваліфікаційна робота у повній мірі відповідає поставленому завданню як в теоретичній, так і в практичній частині

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі висвітлюється актуальність теми роботи, дається аналіз досліджуваної проблеми і обґрунтовується застосовуваний підхід до її вирішення, формулюються цілі і завдання. У першому розділі розглядаються аналізуються підходи до приховування інформації. Наступні розділи присвячені розробці моделі стеганосистеми метод приховування великого об'єму даних в просторовій області зображення Розглянуто питання застосування розробленого методу

4. Позитивні сторони роботи Кваліфікаційна робота містить ряд інноваційних рішень, зокрема, запропонований метод дозволяє приховування великого об'єму даних за рахунок компенсації втрачених бітів під час міжформатного перетворення.

5. Негативні сторони роботи роботи Не проводилося дослідження запропонованого підходу на стійкість до атак на стеганосистему

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення виконане відповідно до теми кваліфікаційної роботи з дотриманням стандартів. В загальному графічне оформлення виконане якісно, пояснювальна записка відповідає нормам щодо її оформлення.

7. Відгук про роботу в цілому В загальному кваліфікаційна робота заслуговує позитивної оцінки. Весь матеріал кваліфікаційної роботи структурований, чіткий та послідовний. Усі розділи роботи послідовні та логічні, що дозволяє чітко розуміти викладений матеріал в рамках тематики кваліфікаційної роботи. Графічний матеріал дозволяє наочно побачити доцільність та ефективність рішень, які були прийняті за основу для досягнення поставленої мети.

8. Інші зауваження Окремі описи в пояснювальній записці подано занадто деталізовано, що ускладнює сприйняття матеріалу фахівцями в обраній предметній галузі.

9. Оцінка кваліфікаційної роботи Враховуючи всі позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінку «добре».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи) Мисан Віктор Володимирович, доцент ТМіТ

«6» 06 2022. (підпис)