МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
Кафедра Комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
Аліна САВЧЕНКО
«_____» _____ 2023р.

КВАЛІФІКАЦІЙНА РОБОТА

(ДИПЛОМНИЙ ПРОЄКТ, ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ **"БАКАЛАВР"**
ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ
"ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ"

Тема: " Застосунок прихованої передачі повідомлень на основі цифрової стеганографії "

Виконавець: Пацанівський Максим Олександрович

Керівник: : к.т.н., доцент Райчев Ігор Едуардович

Нормоконтролер: Олександр ШЕВЧЕНКО

Київ – 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра комп'ютерних інформаційних технологій

Галузь знань, спеціальність: 12 "Інформаційні технології, 122 "Комп'ютерні науки", ОПП "Інформаційні управляючі системи та технології".

ЗАТВЕРДЖУЮ
Завідувач кафедри

Аліна САВЧЕНКО
"____" _____ 2023 р.

ЗАВДАННЯ

на виконання дипломного пректу студентки

Пацанівського Максима Олександровича

1. **Тема роботи:** «Застосунок прихованої передачі повідомлень на основі цифрової стеганографії» затверджена наказом ректора № 623/ст від 01.05.2023.
2. **Термін виконання роботи:** з 15.05.2023р. по 25 .06.2023р.
3. **Вихідні дані до роботи:** алгоритм та компоненти програмного засобу на основі методів стенографії для захисту конфіденційних даних, алгоритм приховування даних методом заміни найменших значущих бітів (LSB) та вилучення даних із утвореного стегоконтейнеру, програмний засіб для захисту даних на основі методів стенографії, який було реалізовано мовою програмування Python.
4. **Зміст пояснювальної записки (перелік питань, що підлягають розробці):** аналіз останніх публікацій, досліджень, алгоритмів по темі стеганографія, реалізація алгоритму Least Significant Bit мовою Python, розробка застосунку з візуалізацією.
5. **Перелік обов'язкового графічного матеріалу:** Слайди презентації MS Powerpoint.

6. Календарний план-графік

№ з/п	*Завдання*	*Термін виконання*	*Підпис керівника*
1.	Дослідження та аналіз предметної області використання	15.05.2023–19.05.2023	
2	Опрацювання інформації за тематикою дипломного проекту	19.05.2023–23.05.2023	
3	Розробка алгоритма та компонентів програмного засобу на основі методів стенографії для захисту конфіденційних даних	21.05.2023–24.05.2023	
4	Розробка алгоритмів приховування даних методом заміни найменших значущих бітів (LSB) та вилучення даних із утвореного стегоконтейнеру	25.05.2023–27.05.2023	
5	Проектування компонентів програмного засобу та реалізація застосунку мовою Python.	28.05.2023 – 02.06.2023	
6	Написання пояснювальної записки дипломного проекту	07.06.2023–15.06.2023	
7	Підготовка демонстраційного матеріалу та доповіді	08.05.2023 – 19.06.2023	

Дата видачі завдання: 15 травня 2023 р.

Керівник дипломного проекту _____ Ігор РАЙЧЕВ

(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Максим ПАЦАНІВСЬКИЙ

(підпис випускника) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до дипломного проєкту «Застосунок прихованої передачі повідомлень на основі цифрової стеганографії» викладена на 73 сторінках, містить 33 рисунків та 18 наукових джерел.

Мета дипломного проекту: з'ясувати практичне значення методів комп'ютерної стеганографії та створити програмний засіб для вбудовування повідомлення до графічних файлів на основі методу заміни найменших значущих бітів, використавши засоби мови програмування Python.

Об'єкт дослідження: стеганографічні методи захисту інформації.

Предмет дослідження: програмна реалізація алгоритму приховування файлів у 24-бітних зображеннях формату bmp засобами Python.

Метод дослідження: аналіз, експеримент.

Результат проекту: спроектований програмний засіб для захисту даних на основі методів стенографії, який було реалізовано мовою програмування Python.

ЦИФРОВА СТЕГАНОГРАФІЯ, АЛГОРИТМИ, ОБРОБКА ЗООБРАЖЕНЬ, СТЕГАНОГРАФІЧНІ МЕТОДИ, АНАЛІЗ, LEAST SIGNIFICANT BIT(LSB), МОДИФІКАЦІЯ ДАНИХ, КРИПТОГРАФІЯ.

ЗМІСТ

ВСТУП 7
РОЗДІЛ 1. МІСЦЕ ЦИФРОВОЇ СТЕГАНОГРАФІЇ СЕРЕД ЗАСОБІВ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ 8
1.1. Способи захисту інформації 8
1.2. Історія розвитку стеганографії 11
1.3. Класифікація стеганографічних методів 13
1.4. Аналіз останніх публікацій, досліджень та існуючих рішень 16
1.5. Роль цифрової стеганографії в сучасному інформаційному суспільстві 18
1.6. Практичні застосування цифрової стеганографії в різних сферах ... 20
1.7. Висновок до розділу 1 22
РОЗДІЛ 2. АЛГОРИТМ ПРИХОВУВАННЯ ПОВІДОМЛЕНЬ МЕТОДОМ ЗАМІНИ НАЙМЕНШИХ ЗНАЧУЩИХ БІТІВ 25
2.1. Принципи роботи методів цифрової стеганографії 25
2.2. Структура графічних файлів формату bmp 29
2.3. Алгоритми приховування файлів методом “заміни наймолодших значущих бітів” та їх відтворення 32
2.4. Алгоритми видобування файлу із стегоконтейнеру 35
2.5. Метод Least Significant Bit 37
2.6. Метод Statistical Analysis 40
2.7. Метод Payload Extraction 43
2.8. Метод Transform Domain Extraction 44
2.9. Метод Neural Network-based Extraction 45
2.10. Висновок до розділу 2 47
РОЗДІЛ 3. ЗАСТОСУВАННЯ ЦИФРОВОЇ СТЕГАНОГРАФІЇ З ВИКОРИСТАННЯМ МОВИ ПРОГРАМУВАННЯ PYTHON 49
3.1. Огляд мови програмування Python та її можливості 49

3.2. Побудова стегоконтейнерів за допомогою зображень у форматі bmp 51
3.3. Реалізація алгоритмів приховування файлів методом "заміни найменших значущих бітів" у мові Python 53
3.4. Реалізація алгоритмів видобування файлів із стегоконтейнерів у мові Python 55
3.5. Використання методу Least Significant Bit у мові Python 56
3.6. Реалізація та демонстрація застосунку на основі цифрової стеганографії. 58
3.7. Висновок до розділу 3 72
ВИСНОВКИ 74
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ. 75

ВСТУП

Актуальність дослідження. У сучасному суспільстві, де комп'ютерні мережі та інформаційні технології постійно розвиваються, інформаційна безпека стає однією з найбільш актуальних проблем. Все більше послуг надається в цифровому форматі, і тому важливо забезпечити надійний захист цієї інформації від різних загроз, таких як несанкціонований доступ, підробка, витік і порушення ліцензійних угод щодо копіювання. Виникнення таких проблем змушує нас удосконалювати методи захисту даних, щоб забезпечити їх ефективність та безпеку. З цим у зв'язку стає необхідним вирішення питань, пов'язаних з розробкою та впровадженням нових технологій та методів захисту інформації, що допоможуть забезпечити її недоступність для несанкціонованого використання і зберегти конфіденційність. Таким чином, актуальність ефективних заходів захисту інформації набуває все більшого значення у сучасному інформаційному суспільстві.

У нашому сучасному суспільстві, яке характеризується швидким розвитком інформаційних технологій, велика увага приділяється забезпеченню конфіденційності та автентичності інформації. Криптографія використовується для захисту цих значущих аспектів інформації і отримала значний розвиток як в Україні, так і в усьому світі. Проте, деякі країни мають обмеження на використання криптографічних засобів. Крім того, існують ситуації, коли важливо приховати навіть сам факт існування конфіденційної інформації, що неможливо досягти за допомогою криптографічних методів. Це стає можливим завдяки цифровій стеганографії, яка дозволяє розв'язати ці завдання. Розвиток комп'ютерної стеганографії є важливим та актуальним напрямом для дослідників України, які працюють у цій науковій галузі. Це дозволить розширити наші знання та розуміння щодо захисту інформації і забезпечення її безпеки.

РОЗДІЛ 1.
МІСЦЕ ЦИФРОВОЇ СТЕГАНОГРАФІЇ СЕРЕД ЗАСОБІВ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

1.1. Способи захисту інформації

Спізнене усвідомлення цінності інформації проявилося ще у минулому, коли важливі листи відомих особистостей ставали об'єктом великого інтересу як з боку їх ворогів, так і з боку союзників. Тоді було поставлене завдання захисту конфіденційності листування від несанкціонованого доступу. У сучасному суспільстві, зростаюча комерційна цінність інформації привела до того, що вона стала надзвичайно поширеним та важливим ресурсом. Інформація виробляється, зберігається, передається, купується й продається, і, відповідно, вона піддається ризику крадіжок та підробок. Отже, безумовно необхідно забезпечувати її надійний захист.

Сучасне суспільство набуває все більшої інформаційної орієнтації, оскільки успіх у різних сферах діяльності залежить від наявності та ефективного використання важливих знань. Цей розвиток природно спричиняє ризики недобросовісного використання інформації та потенційні збитки, пов'язані з несанкціонованим доступом до неї. Відтак, забезпечення захисту інформації стає необхідним і важливим аспектом сучасного суспільства.

На сьогоднішній день сформульовано три базові принципи, яким повинна відповідати інформаційна безпека:

- цілісність даних - це захист від збоїв, що ведуть до втрати інформації, а також захист від неавторизованого створення або знищення даних;

- конфіденційність інформації;

- доступність інформації для всіх авторизованих користувачів.

У сучасному суспільстві зростає потреба в обмеженому доступі до певної інформації. Ця інформація може бути охарактеризована як прихована, приватна, секретна або конфіденційна. Це означає, що лише обмежене коло осіб має право на її доступ. Використовуються різні методи та підходи, щоб забезпечити цю конфіденційність та зберегти інформацію від несанкціонованого доступу. Забезпечення безпеки конфіденційної інформації стає дедалі важливішою проблемою в нашому інформаційному суспільстві, а розвиток ефективних методів захисту і збереження такої інформації є актуальним завданням для науковців та дослідників.Наприклад, мова може йти про медичну таємницю, юридичну, комерційну, військову. І в залежності від шкоди через розголошення інформації, їй може надаватись статус державної таємниці.

На сьогоднішній день існує багато різних засобів захисту даних, але найпоширенішими є такі:

1. Криптографічні методи захисту даних: це включає в себе шифрування повідомлень та інформації з використанням різних криптографічних алгоритмів, таких як AES, RSA, і ECC. Ці методи захисту даних забезпечують конфіденційність та цілісність даних, тобто унеможливлюють зламати шифр.

2. Автентифікація та авторизація: ці методи захисту даних дозволяють перевірити, що користувач, який намагається отримати доступ до системи або даних, дійсно має на це право. Автентифікація може бути здійснена за допомогою паролів, біометричних даних (відбитків пальців, скану очей тощо) або інших методів. Авторизація забезпечує контроль доступу до системи або даних на основі різних рівнів дозволів.

3. Файерволи: це програмне забезпечення, яке перевіряє вхідні та вихідні пакети даних на наявність шкідливого вмісту та блокує їх, якщо потрібно. Файерволи забезпечують безпеку мережі, блокуючи віруси та інші види атак.

4. Перевірка безпеки веб-додатків: це методи захисту даних, які використовуються для перевірки веб-додатків на вразливості та можливість атаки.

Це може включати тестування на проникнення, аудит коду та перевірку на відповідність стандартам безпеки.

5. Захист від соціальної інженерії: це методи захисту даних, які дозволяють запобігти атакам з використанням соціальної інженерії, таких як фішинг, імітація довіреності, підманювання тощо. Ці методи включають в себе освіту користувачів, перевірку електронної пошти та інших повідомлень на підозрілий вміст, та застосування фільтрів спаму.

6. Заборона використання простих паролів: цей метод захисту даних рекомендує використовувати складні паролі, які складаються з різних символів та чисел. Такі паролі унеможливлюють зламати шифр методом перебору.

7. Захист фізичного доступу: це методи захисту даних, які включають у себе контроль за фізичним доступом до комп'ютерів, пристроїв зберігання даних та інших об'єктів. Це може включати в себе застосування системи контролю доступу та камер відеоспостереження.

8. Спосіб приховати передачу інформації можна здійснити за допомогою стеганографії. Цей метод дозволяє приховати факт передачі інформації в іншому об'єкті, такому як зображення, без виклику підозрілості.

Сьогодні нерідко виникає необхідність передати конфіденційне повідомлення невеликого обсягу, при цьому використання складних криптографічних систем за рядом причин неможливе. Виходом з цієї ситуації є використання стеганографії. Для передачі повідомлення його елементи вписуються до звукового, графічного чи відео файлу. При цьому в початковому файлі деякі байти змінюються, але в цілому запідозрити, що в картинці чи аудіо файлі знаходиться деяке повідомлення, дуже складно. Тому передається начебто звичайний файл, а насправді там знаходиться деяке секретне повідомлення.

1.2. Історія розвитку стеганографії

Стеганографія - це наука, яка досліджує методики та технології прихованої передачі інформації шляхом збереження таємниці самого факту передачі. За допомогою стеганографії інформація може бути вбудована в різні типи даних, такі як зображення, аудіо та відео, зберігаючи тим самим конфіденційність інформації. Це дає змогу забезпечити безпеку інформації при її передачі та зберіганні, тим самим захищаючи її від неповажного використання та несанкціонованого доступу.

Стеганографія як наука стала відома громадськості лише в останні десятиріччя, хоча методи приховування інформації разом із шифрувальною справою (криптографією) відомі ще із часів Давнього світу (див. рис.1.1).

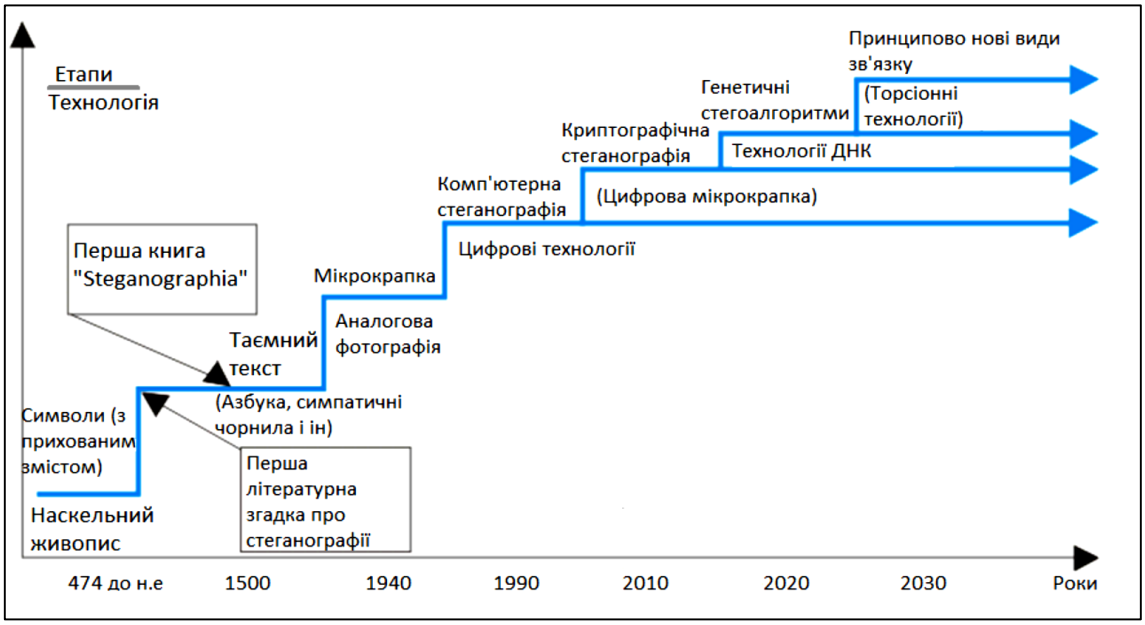


Рис.1.1. Історія розвитку стеганографії

Про перший загальновідомий факт застосування стеганографії «на ділі» стало відомо від американських спецслужб після терактів 11 вересня 2001 року. Тоді терористи в процесі організації викрадення пасажирських літаків використовували стеганографії для прихованого обміну інформацією між собою.

Багато дослідників вважають, що Єгипет є місцем народження стеганографії. Однак, є також вказівки на те, що першими формами "стеганографічних

повідомлень" можуть вважатися наскельні малюнки, створені давніми людьми. Ці ранні форми стеганографії є доказом існування інтересу до приховування інформації та передачі її тільки обмеженому колу осіб.

Історично цей напрям з'явився першим, але потім багато в чому був витисненим криптографією. Спільна риса цих способів та, що приховуване повідомлення вбудовується в певний об'єкт, що не привертає увагу. Далі цей об'єкт відкрито транспортується адресату. При криптографії наявність шифрованого повідомлення сама собою привертає увагу зловмисників, при стеганографії ж наявність прихованих даних залишається непомітною.

Перший запис про використання стеганографії зустрічається в трактаті Геродота «Історія», що відноситься до 440 року до н. е. У трактаті були описані два методи приховування інформації. Демарат відправив попередження про майбутній напад на Грецію, записавши його на дерев'яну підкладку воскової таблички до нанесення воску. Другий спосіб полягав у наступному: на поголену голову раба записувалося необхідне повідомлення, а коли його волосся відростало, він вирушав до адресата, який знову голив його голову і зчитував доставлене повідомлення.

Ще древні римляни писали між рядків невидимим чорнилом, у якості яких використовувалися фруктові соки, сеча, молоко і деякі інші натуральні речовини. Текст, записаний такими чорнилом, проявляється лише за певних умов (нагрівання, освітлення, хімічний проявник і т. д.).

У XV столітті чернець Трітеміус (1462-1516), який займався криптографією і стеганографією, описав багато різних методів прихованої передачі повідомлень. Пізніше, в 1499 році, ці записи були об'єднані в книгу «Steganographia».

Комп'ютерні технології додали новий імпульс розвитку й удосконалюванню стеганографії, з'явився новий напрямок в області захисту інформації - комп'ютерна стеганографія. Існують два ключових напрямки використання комп`ютерної стеганографії: пов'язаний з цифровою обробкою сигналів і не пов'язаний. У першому випадку секретні повідомлення вбудовуються у цифрові дані, які як правило, мають аналогову природу. У другому — конфіденційна інформація розміщується в заголовках файлів чи пакетів даних. Однак цей напрямок не знайшов

широкого застосування через відносну легкість розкриття та знищення прихованої інформації.

Не зважаючи на численні відкриті публікації та щорічні конференції, тривалий час стеганографія не мала усталеної термінології. Основні поняття стеганографії були узгоджені у 1996 р. на 1-й Міжнародній конференції з приховування даних — Information Workshop on Information Hiding'96.

1.3. Класифікація стеганографічних методів

Стеганографію можна класифікувати за методами, які використовуються для приховання інформації. Технологічні методи включають хімічні та фізичні техніки. Хімічні методи використовують органічні речовини та невидимі чорнила для приховання інформації. Фізичні методи включають застосування схованок, мікрокрапок, камуфляжу та голограм для приховання інформації. Схованки передбачають використання таємних приміщень для зберігання інформації. Мікрокрапки використовуються для приховання інформації у текстових документах за допомогою дуже маленьких пробілів між словами. Камуфляж передбачає розміщення інформації в середині існуючих об'єктів, таких як зображення, з метою зробити їх непомітними для зовнішнього спостереження. Голограми використовуються для захисту від підробки та включають у себе складні оптичні ефекти, які дозволяють використовувати двовимірні зображення для приховання тривимірної інформації.

Наведена таблиця 1.1 порівняння різних методів стеганографії. Ця таблиця допоможе зрозуміти особливості кожного методу

Таблиця 1.1

Основні методи стеганографії та їх опис

Метод стеганографії	Опис
LSB-метод	Цей метод полягає у внесенні прихованого повідомлення в найменш значущі біти пікселів зображення.
Частотний аналіз	Цей метод використовує спектральні характеристики зображення для приховання повідомлення.
Алгоритми заміни	Цей метод замінює певні пікселі або регіони зображення для приховування повідомлення.
Алгоритми перестановки	Цей метод переставляє пікселі або блоки пікселів у зображенні для приховування повідомлення.

У таблиці 1.2 розглянемо переваги та недоліки основних методів стеганографії. Використовуючи таблицю показану нижче ми можемо проаналізувати та порівняти різні підходи до прихованої передачі повідомлень, щоб більш детально проаналізувати їх різноманітність та ефективність.

Таблиця порівняння методів стеганографії пропонує глибокий огляд різних підходів до прихованої передачі повідомлень, що дозволяє дослідникам і професіоналам зробити обґрунтований вибір для своїх конкретних потреб. Кожен метод в таблиці має свої унікальні переваги та недоліки, які слід врахувати під час вибору оптимального рішення.

Таблиця 1.2

Переваги та недоліки основних методів стеганографії

Метод стеганографії	Переваги	Недоліки
LSB-метод	Простий у реалізації, підходить для різних форматів зображень, велика ємність для прихованого повідомлення.	Може бути вразливий до атак, що змінюють малозначущі біти зображення, може призводити до втрати якості зображення.
Частотний аналіз	Може бути ефективним для стеганографії звуку та відео, може забезпечити високу помітність прихованого повідомлення.	Вимагає великої обчислювальної потужності, може бути вразливий до атак на спектральну область зображення.
Алгоритми заміни	Забезпечує високу стійкість до атак на найменш значущі біти, може бути використаний для приховування багатошарових повідомлень.	Може призводити до помітних змін в зображенні, вимагає точного моделювання регіонів заміни.
Алгоритми перестановки	Забезпечує високий рівень стійкості до атак на пікселі та регіони, може забезпечити хорошу помітність для прихованого повідомлення, може бути використаний для створення складних стеганографічних схем.	Вимагає великої обчислювальної потужності, може призводити до зміни глобальної структури зображення.

Порівняння методів стеганографії допомагає визначити найбільш підходящий варіант для конкретних сценаріїв використання. Крім унікальних переваг кожного методу, також варто враховувати наступні аспекти:

- Ступінь помітності: Важливо оцінити, наскільки помітне є вбудоване повідомлення. Деякі методи можуть залишати сліди, що робить їх менш захищеними.

- Обмеження обсягу повідомлення: Деякі методи можуть мати обмежену ємність для вбудовування повідомлень. Важливо переконатися, що обраний метод може вмістити необхідний обсяг інформації.
- Відновлюваність: Важливо мати можливість відновити повідомлення в разі його пошкодження або втрати. Деякі методи можуть мати механізми відновлення, що дозволяють відновити частину втрачених даних.
- Вплив на оригінальний контент: Деякі методи можуть внести зміни в оригінальні дані, що може вплинути на їх якість або цілісність. Важливо врахувати, наскільки важливе збереження оригінального вмісту.
- Стійкість до атак: Різні методи можуть мати різні рівні стійкості до різних типів атак, таких як статистичний аналіз, знаходження зразків або обчислювальні методи. Важливо оцінити, наскільки захищений обраний метод проти потенційних загроз.

1.4. Аналіз останніх публікацій, досліджень та існуючих рішень

Предметом вивчення комп'ютерної стеганографії є такі технології, які приховують інформацію у потоках оцифрованих сигналів та реалізуються на базі комп'ютерної техніки і програмного забезпечення в рамках окремих обчислювальних систем, корпоративних чи глобальних мереж. Ця наука інтегрує в собі здобутки криптографії, теорії інформації, теорії ймовірності та математичної статистики, теорії дискретних ортогональних перетворень, цифрової обробки сигналів та зображень, розпізнавання образів та ін.

Одним з основних завдань комп'ютерної стеганографії є забезпечення конфіденційності передачі інформації між сторонами, які не бажають, щоб присутність такої інформації стала відомою третім особам. Крім того, стеганографічні технології можуть використовуватись для різних цілей, таких як забезпечення автентичності та цілісності даних, зменшення шуму та покращення якості передачі сигналу. Важливою частиною комп'ютерної стеганографії є аналіз методів та атак, що застосовуються для виявлення прихованої інформації, та

розробка заходів для захисту від таких атак. Розвиток комп'ютерної стеганографії відкриває широкі можливості для захисту конфіденційності інформації в цифровому світі, а також для застосування в таких галузях, як криміналістика, наукові дослідження, оборона та багато інших.

У розвиток вітчизняної стеганографії зробили вагомий внесок М.Є.Шелест, Г.Ф.Конахович, А.Ю.Пузиренко, В.О.Хорошко та інші учені і практики, який полягає у розробці та удосконаленні моделей, методів та засобів стеганографії і стеганоаналізу, оцінюванні їх характеристик тощо.

Сьогодні не бракує стеганографічних програм як початкового, так і професійного рівня, але захищеність їх коду не дозволяє простежити методи, закладені в основу алгоритмів їх дії. Реалізовані програмні засоби приховування інформації, наприклад, Steganos, Outguess, Jsteg, Steghide, Jphs, S-Tools та ін. є нескладними у використанні та здатні створити стеганоканал з високою пропускною здатністю.

Незважаючи на наявність багатьох стеганографічних програм, захист від їх виявлення та розшифрування залишається одним із основних завдань цієї науки. Один із підходів до розв'язання цього завдання полягає в розробці нових методів та алгоритмів, які були б нечутними для сучасних засобів аналізу та розшифрування інформації. Для цього використовуються різні техніки, такі як аналіз чутливості стеганоканалу до стеганалізу, розвиток нових криптографічних методів, створення нових методів забезпечення конфіденційності та стійкості до атак. Крім того, важливим є підтримка наукової спільноти та обмін досвідом та знаннями між вченими та спеціалістами у цій області. Тільки таким чином можна забезпечити стійкість та ефективність стеганографічних систем в майбутньому.

Одним з перспективних і актуальних напрямків є побудова стеганографічних систем на основі взаємодії криптографії і стеганографії, коли, з одного боку, застосовуються перевірені на стійкість криптографічні алгоритми, а з іншого, стійкі стеганографічні алгоритми, що відповідають певним вимогам і обмеженням, а також правильного їх узгодження.

Отже, актуальність питання інформаційної безпеки постійно зростає в зв’язку зі збільшенням кількості даних. У сфері інформаційної безпеки є два основних напрямки: криптографія і стеганографія. Широке застосування методів прихованої передачі даних за допомогою стеганографії призвело до розвитку її методів. В даний час комп'ютерна стеганографія продовжує розвиватися:

- формується теоретична база;
- ведеться розробка нових, більш стійких методів вбудовування повідомлень.

Серед основних причин спостереження сплеску інтересу до стеганографії можна виділити прийняті в ряді країн обмеження на використання сильної криптографії, а також проблему захисту авторських прав на художні твори в цифрових глобальних мережах. Тому найближчим часом можна чекати нових публікацій та розробок у цій області.

1.5. Роль цифрової стеганографії в сучасному інформаційному суспільстві

Цифрова стеганографія відіграє надзвичайно важливу роль у сучасному інформаційному суспільстві. З розвитком технологій та зростанням кількості передаваної інформації, забезпечення конфіденційності та безпеки стало критично важливою задачею. Цифрова стеганографія надає можливість приховувати інформацію в беззагальному вигляді, так що її наявність стає непомітною для недозволених осіб.

Перший аспект ролі цифрової стеганографії полягає в забезпеченні конфіденційності даних. Вона дозволяє передавати повідомлення чи файли, приховуючи їх у звичайних носіях, таких як зображення, звуки або відеофайли. Це забезпечує високий рівень безпеки, оскільки зовнішній спостерігач не може виявити приховану інформацію, навіть якщо отримує доступ до носія.

Другий аспект ролі цифрової стеганографії полягає в захисті від виявлення самого факту існування конфіденційної інформації. У деяких випадках важливо не тільки заховати саме повідомлення, але й приховати той факт, що щось приховане

існує. Це особливо важливо для ситуацій, коли доступ до інформації заборонений або небажаний.

Третій аспект ролі цифрової стеганографії пов'язаний з автентичністю даних. Вона може бути використана для підтвердження, що дані не були змінені під час передачі. Шляхом приховування контрольних сум або хеш-значень в стеганографічних носіях, можна забезпечити цілісність і недоторканість даних. Такий підхід дозволяє виявити будь-які зміни в прихованій інформації та відокремити автентичні дані від потенційно підроблених.

Четвертий аспект ролі цифрової стеганографії пов'язаний з боротьбою зі шпигунством та кібератаками. Стеганографія може бути використана для захисту важливої інформації від зловмисників, які намагаються перехопити комунікацію або отримати несанкціонований доступ до даних. Приховання інформації може унеможливити її виявлення та зрозуміння зловмисниками, забезпечуючи вищий рівень безпеки.

П'ятий аспект ролі цифрової стеганографії полягає в захисті особистої приватності та конфіденційності користувачів. Вона дозволяє зберегти особисті дані, такі як паролі, персональну інформацію або фінансові дані, у прихованих форматах, зменшуючи ризик їх незаконного доступу чи зловживання.

Шостий аспект ролі цифрової стеганографії пов'язаний зі створенням та збереженням архівів та резервних копій даних. Приховання важливих даних в звичайних носіях дозволяє створювати непомітні архіви, що підвищує їх захищеність та забезпечує доступ до них лише авторизованим користувачам.

Сьомий аспект ролі цифрової стеганографії полягає в захисті комерційної та корпоративної інформації. Компанії та бізнеси можуть використовувати стеганографію для збереження конфіденційності своїх комерційних секретів, технологій, стратегій розвитку та іншої конфіденційної інформації. Це надає їм перевагу на ринку, оскільки їхні конкуренти не можуть отримати доступ до цієї цінної інформації.

Восьмий аспект ролі цифрової стеганографії пов'язаний з правоохоронною діяльністю та розслідуваннями. Люди, які займаються боротьбою зі злочинністю та

тероризмом, можуть використовувати стеганографію для обміну конфіденційною інформацією, яка не підлягає виявленню зловмисниками. Це допомагає забезпечувати безпеку та ефективність правоохоронних органів.

Дев'ятий аспект ролі цифрової стеганографії полягає в захисті авторських прав та інтелектуальної власності. Автори, художники, музиканти та інші творчі люди можуть використовувати стеганографію для прихованого вбудовування своїх підписів, маркерів або захисту від підробки своїх творінь. Це дозволяє їм захистити свої права та визначити свою авторську належність.

Десятий аспект ролі цифрової стеганографії пов'язаний зі забезпеченням надійної та безпечної комунікації. В сфері дипломатії, військових операцій, ділового спілкування та особистих переговорів стеганографія може забезпечити безпечний обмін інформацією між сторонами, знижуючи ризик перехоплення та недозволеного доступу до даних.

Загалом, роль цифрової стеганографії в сучасному інформаційному суспільстві полягає в забезпеченні конфіденційності, захисті від виявлення, автентичності даних, боротьбі зі шпигунством та кібератаками, захисті особистої приватності, створенні архівів та резервних копій даних, захисті комерційної та корпоративної інформації, сприянні правоохоронній діяльності, захисті авторських прав та інтелектуальної власності та забезпеченні безпечної комунікації.

1.6. Практичні застосування цифрової стеганографії в різних сферах

Виклики та перспективи розвитку цифрової стеганографії є актуальною темою в сучасному інформаційному суспільстві. Зростання обсягу цифрових даних та швидкість передачі інформації ставлять перед стеганографією нові виклики і вимагають постійного розвитку та вдосконалення методів.

Одним з основних викликів є розвиток сучасних технологій обробки та аналізу даних. З появою штучного інтелекту, машинного навчання та глибокого навчання, необхідно розробляти стеганографічні методи, які можуть стійко працювати у таких умовах та уникати виявлення.

Інший виклик полягає в адаптації до зростаючої кількості цифрових каналів передачі даних. З поширенням мобільних та бездротових технологій з'являються нові можливості для передачі інформації, але це також вимагає розробки ефективних методів стеганографії, які можуть працювати на різних типах каналів.

Також важливим викликом є боротьба з сучасними методами криптоаналізу. Залежно від рівня складності алгоритмів стеганографії, їх можуть аналізувати й розкривати зловмисники або комп'ютерні програми. Тому необхідно постійно вдосконалювати стеганографічні методи, щоб забезпечити стійкість до криптоаналітичних атак.

Значний вплив на розвиток цифрової стеганографії мають також правові та етичні аспекти. Багато країн мають законодавство, що регулює використання стеганографії, особливо в контексті боротьби зі злочинністю та тероризмом. Це ставить виклик перед дослідниками та розробниками, які повинні забезпечити баланс між захистом приватності та безпеки і несанкціонованим використанням стеганографії.

Однак, разом з викликами, у цифровій стеганографії є і перспективи розвитку. З появою нових технологій, таких як квантові обчислення та блокчейн, відкриваються нові можливості для розробки стійких та надійних методів стеганографії, які можуть забезпечити високий рівень безпеки передачі та збереження конфіденційної інформації.

Розробка стеганографії на основі штучного інтелекту є однією з перспективних напрямків. Використання алгоритмів машинного навчання та глибокого навчання може дозволити розробити більш ефективні та незламні методи стеганографії, здатні використовувати складні шаблони та приховуватися від детекторів.

Розвиток стеганографії також пов'язаний з розширенням її застосувань. Наприклад, використання стеганографії у сфері медицини може допомогти забезпечити конфіденційність та безпеку обміну медичними даними пацієнтів. Також використання стеганографії у сфері медіа може дозволити авторам та

виробникам контенту захистити свої права та запобігти підробці або незаконному розповсюдженню.

Перспективою розвитку цифрової стеганографії є також її поєднання з іншими технологіями, наприклад, квантовою криптографією. Квантові методи можуть забезпечити неперебірну безпеку комунікації, а стеганографія може використовуватися для приховування квантових ключів та забезпечення безпеки передачі.

Загалом, цифрова стеганографія відіграє важливу роль у сучасному інформаційному суспільстві, забезпечуючи конфіденційність, захист від виявлення та цілісність даних. Вона стикається зі значними викликами, пов'язаними з технологічними, правовими та етичними аспектами, але також має перспективи розвитку, які пов'язані з використанням нових технологій та розширенням її застосувань у різних сферах. Дослідники та розробники продовжують працювати над вдосконаленням методів стеганографії, щоб забезпечити безпеку та приватність у цифровому світі.

1.7. Висновок до розділу 1

У даному розділі були розглянуті та проаналізовані різні аспекти цифрової стеганографії, її роль у сучасному інформаційному суспільстві, виклики та перспективи її розвитку. Також висвітлено основні методи захисту інформації за допомогою стеганографії, такі як вбудовування даних у носії, шифрування та аутентифікація. Було показано, що ці методи є ефективними для забезпечення конфіденційності та цілісності даних.

Вивчення історії розвитку стеганографії розкриває, що ця наука виникла ще в стародавні часи і постійно знаходила своє застосування для приховування повідомлень. З того часу стеганографія пройшла значний шлях розвитку, а її методи та технології постійно еволюціонують. Використовуючи різні техніки та носії інформації, вона стала надійним засобом передачі конфіденційних даних в сучасному інформаційному суспільстві.

У дослідженні класифікації стеганографічних методів були визначені основні підходи до класифікації цих методів. Дослідники розрізнили різні типи стеганографічних методів, включаючи ті, що базуються на зміні пікселів, зміні частоти аудіосигналів та текстових методах. Виявлено, що різноманітність цих методів дозволяє успішно приховувати дані в різних типах носіїв інформації. Такий підхід до класифікації сприяє розумінню і використанню різноманітних стеганографічних методів для забезпечення конфіденційності інформації.

У процесі аналізу останніх публікацій, досліджень та існуючих рішень були виявлені актуальні тенденції у галузі цифрової стеганографії. Дослідники систематично працюють над вдосконаленням і розробкою нових аспектів цифрової стеганографії. У результаті аналізу були відзначені нові алгоритми та методи вбудовування даних, а також дослідження, спрямовані на виявлення та аналіз стеганографічних атак. Це свідчить про постійний інтерес до розвитку цифрової стеганографії та важливість подальшої роботи в цьому напрямку. Нові дослідження сприятимуть розширенню можливостей стеганографії та забезпеченню більш ефективного захисту конфіденційної інформації.

На основі проведеного дослідження можна зробити висновок, що цифрова стеганографія відіграє важливу роль у сучасному інформаційному суспільстві. Вона дозволяє забезпечити конфіденційність та цілісність інформації, що передається через різні канали зв'язку. Стеганографічні методи можуть бути використані в різних сферах, включаючи комунікації, захист даних, криміналістичні дослідження та військову діяльність.

Однак, разом з тим, цифрова стеганографія стикається з викликами і проблемами. З одного боку, постійний розвиток технологій призводить до появи нових методів атак і способів виявлення стеганографічних даних. Це вимагає постійного вдосконалення і адаптації стеганографічних методів. З іншого боку, виникають етичні та правові питання щодо використання стеганографії, зокрема у злочинній та терористичній діяльності.

На основі виконаного аналізу була сформована мета дослідження кваліфікаційної роботи, а саме: з’ясувати практичне значення методів комп’ютерної стеганографії.

РОЗДІЛ 2.
АЛГОРИТМ ПРИХОВУВАННЯ ПОВІДОМЛЕНЬ МЕТОДОМ ЗАМІНИ НАЙМЕНШИХ ЗНАЧУЩИХ БІТІВ

2.1. Принципи роботи методів цифрової стеганографії

Розвиток сучасних цифрових технологій стимулював прогрес у галузі цифрової стеганографії, яка використовується для приховування секретних повідомлень у цифрових даних, таких як зображення, аудіозаписи та відео. Ця технологія дозволяє вбудовувати інформацію в файлові формати, які зазвичай мають аналогову природу. Вбудовування секретних даних може відбуватися не лише у звичайні файли, але й у текстові та скомпресовані формати, розширюючи можливості стеганографічних методів у цифровому середовищі. Узагальнена модель стеганографії рис.2.1.

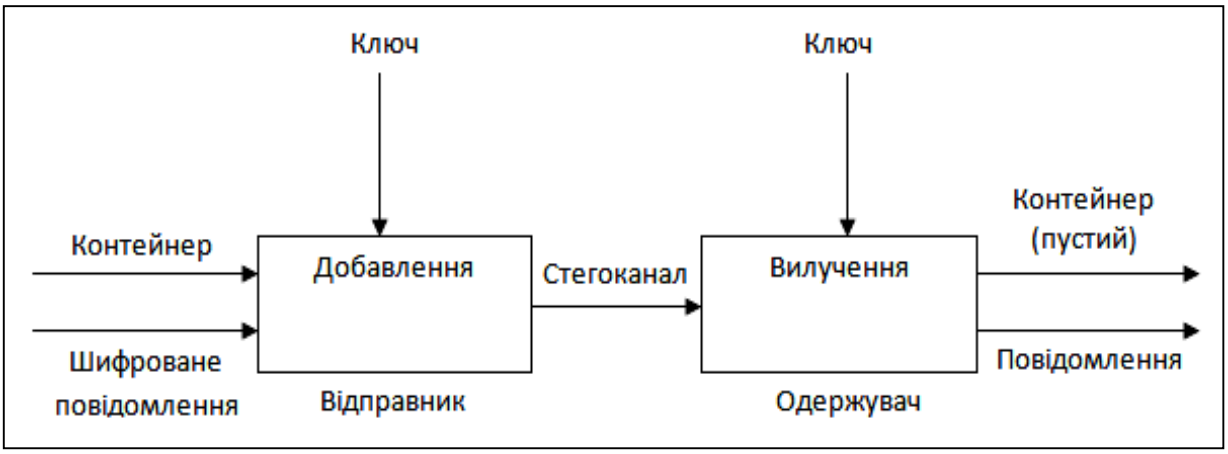


Рис. 2.1. Узагальнена модель стеганографії

Розширений доступ до потужних обчислювальних ресурсів та збільшення обсягу доступної інформації поклали основи для подальшого розвитку цифрової стеганографії. Цей підхід до приховування повідомлень відіграє важливу роль у забезпеченні безпеки даних, конфіденційності та захисту інформаційних потоків.

Дослідження в галузі аналізу ефективності та стійкості методу "заміни найменших значущих бітів" є актуальними в контексті забезпечення надійного збереження секретних даних у цифровому середовищі.

Розуміння переваг і обмежень цього методу дозволяє вдосконалювати алгоритми стеганографії та розробляти ефективні заходи для протидії можливим атакам та виявленню прихованої інформації.

Багато з методів комп'ютерної стеганографії базуються на двох основних принципах, які відкривають широкі можливості для приховування повідомлень у цифрових файлах.

По-перше, файли, які не вимагають абсолютної точності, можуть бути відтворені з певним ступенем змін без впливу на їхню сприйнятливість або функціональність. Це дозволяє нам використовувати їх як "несумісні місця" для приховування додаткової інформації, не спричиняючи підозрілості.

По-друге, наша спроможність розрізнити незначні зміни в таких файлах є обмеженою, особливо без спеціальних інструментів аналізу. Це важливо для забезпечення незловживання та захисту прихованої інформації.

Комбінація цих принципів відкриває шляхи для створення ефективних алгоритмів стеганографії та забезпечення стійкості й невиявленості прихованої інформації.В даний час існує досить багато різних методів (і їх варіантів) вбудовування повідомлень (мається на увазі і вбудовування цифрових водяних знаків).

Метод LSB (Least Significant Bit) Extraction є одним з ключових алгоритмів, використовуваних для вилучення конфіденційної інформації зі стеганографічних зображень. Цей метод ґрунтується на особливостях кодування інформації в найменш значущих бітах пікселів. Найменш значущий біт є найменш вагомим бітом

в числовому представленні пікселя і має мінімальний вплив на візуальне сприйняття зображення.

Процес вилучення прихованої інформації за допомогою методу LSB Extraction включає проходження через кожен піксель зображення та витягнення його найменш значущого біта. Ці найменш значущі біти потім збираються разом для відновлення початкового повідомлення, яке було приховане у зображенні. Цей процес дозволяє відновити приховану інформацію без значних втрат або спотворень зображення. Statistical Analysis (Статистичний аналіз): Цей алгоритм базується на аналізі статистичних властивостей стегоконтейнера. Він виявляє відхилення від очікуваного розподілу пікселів чи байтів у зображенні, що може свідчити про наявність прихованої інформації. Алгоритм використовує статистичні методи для виявлення та видобування цих відхилень.

Payload Extraction (Вилучення за допомогою навантаження): Цей алгоритм базується на структурі стегоконтейнера та специфічних ознаках, які використовуються для вбудовування прихованої інформації. Він аналізує структуру файлу, виявляє маркери або особливості, що вказують на наявність прихованої інформації, і видобуває цю інформацію згідно зі специфікаціями стеганографічного методу.

Transform Domain Extraction (Вилучення в області перетворення): Цей алгоритм використовує перетворення, такі як Дискретне Косинусне Перетворення (DCT) або Вейвлет-перетворення, для видобування прихованої інформації зі стегоконтейнера. Він використовує особливості спектрального представлення даних та виявляє зміни, що відбулися в результаті вбудовування повідомлення.

Neural Network-based Extraction (Вилучення на основі нейронних мереж): З використанням розширення застосування штучних нейронних мереж до області стеганографії, розроблені методи, які використовують нейронні мережі для видобування прихованої інформації. Ці методи можуть аналізувати структуру та характеристики стегоконтейнера, щоб визначити наявність та видобути приховане повідомлення.

Найбільш поширеним є метод заміни найменших значущих бітів або LSB-метод. Суть цього методу полягає в заміні останніх значущих бітів в контейнері (зображенні, аудіо або відеозапису) на біти приховуваного повідомлення. Різниця між порожнім і заповненим контейнерами повинна бути не відчутна для органів сприйняття людини.

Однак, важливо розуміти, що ефективність та стійкість методу "заміни найменших значущих бітів" може залежати від різних факторів, включаючи формат зображення, рівень компресії файлу та характеристики алгоритму стеганографії. Деякі формати зображень можуть бути більш вразливими до виявлення або втрати прихованої інформації, ніж інші. Також варто враховувати, що зростання розміру прихованого повідомлення може призвести до більш помітних змін в стегоконтейнері. Тому, аналіз впливу формату зображення на ефективність та стійкість методу "заміни найменших значущих бітів" є важливим аспектом досліджень у цій галузі. Розуміння цих впливів дозволяє вдосконалювати методи стеганографії та розробляти оптимальні стратегії для забезпечення надійного та невиявного приховування повідомлень.

Більшість досліджень присвячено використанню в якості стеганоконтейнерів саме зображень рис.2.2.

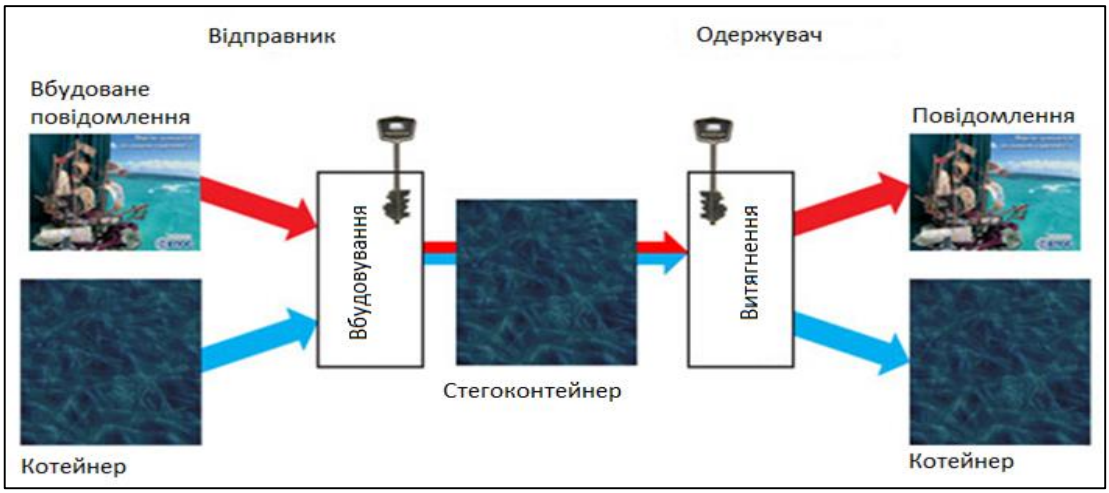


Рис.2.2. Зображення у якості стегоконтейнера

Це обумовлено наступними причинами:

• існуванням практичної необхідності захисту цифрових фотографій, зображень, відео від протизаконного тиражування і розповсюдження;

• заздалегідь відомим (фіксованим) розміром контейнера, відсутністю обмежень, які накладаються вимогами приховування в реальному часі;

• наявністю в більшості реальних зображень текстурних областей, що мають шумову структуру і які найкраще підходять для вбудовування інформації;

• малою чутливістю людського ока до незначних змін кольорів зображення, його яскравості, контрастності, вмісту в ньому шуму, спотворень поблизу контурів.

2.2. Структура графічних файлів формату bmp

Структура графічних файлів формату BMP (Bitmap) є одним із найпоширеніших форматів для збереження растрових зображень. Вона використовується у багатьох програмах і системах, що працюють з графікою. Формат BMP був розроблений компанією Microsoft і має деякі характеристики, які його відрізняють від інших форматів зображень.

Основною особливістю формату BMP є те, що він зберігає кожен піксель зображення окремо, включаючи його колір, положення і інші атрибути. Кожен піксель кодується з використанням бітового представлення, що дозволяє точно відтворити оригінальне зображення. Такий піксельний підхід забезпечує високу якість і точність зображення, але водночас призводить до зайнятості значної кількості дискового простору.

Структура файлів формату BMP включає заголовок, що містить інформацію про розмір файлу, формат зображення, кількість біт на піксель, розташування даних та інші параметри. Після заголовка слідують додаткові блоки, що містять палітру кольорів, дані пікселів та можливі метадані.

Ця структура дозволяє зберігати різноманітні дані в графічних файлах формату BMP, такі як зображення з великою глибиною кольору, альфа-канали для прозорості, маски, градієнти та інші ефекти. Формат BMP також підтримує

стиснення даних для зменшення розміру файлу, але це може призводити до втрати якості зображення.

Загальна структура графічних файлів формату BMP визначена специфікацією і може варіюватись залежно від версії формату та конкретних вимог програми чи системи, що створює або обробляє BMP файли. Однак, незалежно від версії, заголовок файлу завжди містить певні обов'язкові поля, такі як тип файлу, розмір файлу, розміри зображення, кількість біт на піксель та деякі контрольні суми.

У форматі BMP можуть використовуватись різні варіації колірної моделі, такі як RGB (Red, Green, Blue), CMYK (Cyan, Magenta, Yellow, Black), відтінки сірого тощо. Це дозволяє зберігати зображення з різними колірними просторами, що використовуються у візуальних програмах та друкованих матеріалах.

Загалом, структура графічних файлів формату BMP дозволяє зберігати різноманітні дані із зображеннями, забезпечуючи точність, колірну глибину, масштабованість та інші характеристики. Це робить його важливим форматом для роботи з графікою в різних програмах та системах.

Файл bmp завжди складається з 4 частин: файлового заголовка, інформації про зображення та палітру, і блок даних рис.2.3.

Рис.2.3. Структура файлу bmp формату

У форматі BMP графічне зображення зберігається у вигляді матриці значень відтінків кольору для кожної точки. Кожен піксель кодується за допомогою трьох відтінків RGB (Red, Green, Blue). Кожен з цих відтінків представлений числом від 0 до 255 і займає 1 байт або 8 бітів. Таким чином, на кожну точку зображення відводиться 24 біти або 3 байти. рис.2.4.

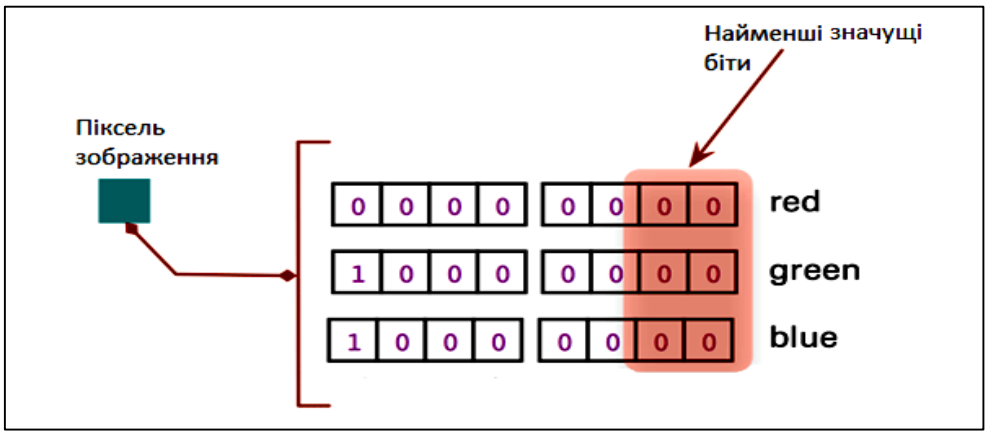


Рис.2.4. Формування кольору точки

Колір кожного пікселя форматується у форматі RGB, де кожен відтінок визначається своїм числовим значенням. Наприклад, червоний колір може мати значення (255, 0, 0), де перший компонент відповідає червоному відтінку, другий - зеленому, а третій - синьому. Змішування цих трьох відтінків в різних пропорціях дозволяє отримати широкий спектр кольорів.

Для 24-х бітного кольору зміна в кожному з трьох каналів одного найменш значимого біта призводить до зміни менш ніж на 1% інтенсивності даної точки, що дозволяє змінювати їх непомітно для ока людини на свій розсуд.

Одна з особливостей BMP формату - підтримка безстислових та стислових зображень. Безстислове зображення зберігає дані про кожен піксель окремо, що забезпечує максимальну точність, але вимагає більшого обсягу дискового простору. Стислі зображення використовують різні алгоритми стиснення даних для зменшення розміру файлу, але при цьому може відбуватись деяка втрата якості.

Крім того, BMP формат дозволяє вбудовувати додаткові метадані в файл, такі як автор, дата створення, коментарі та інші інформаційні поля. Це робить його зручним для зберігання і передачі додаткової інформації разом з зображенням.

2.3. Алгоритми приховування файлів методом “заміни наймолодших значущих бітів” та їх відтворення

Алгоритм "заміни наймолодших значущих бітів" (англ. Least Significant Bit, LSB) є одним з найпоширеніших методів приховування файлів у стеганографії. Тому розглянемо процес приховування і відтворення файлів за допомогою цього методу.

Приховування файлу методом LSB:

1. Виберіть файл-носій (контейнер), який буде використовуватись для приховування. Зазвичай це є зображення, звуковий або відео файл;
2. Оберіть файл, який ви хочете приховати (повідомлення);
3. Перетворіть повідомлення у бінарний формат, розбивши його на окремі біти;
4. Замініть наймолодші значущі біти в контейнері на біти повідомлення. Наприклад, якщо контейнер має піксельну структуру, то замініть наймолодші значущі біти кожного пікселя на біти повідомлення.

Відтворення файлу з методом LSB:

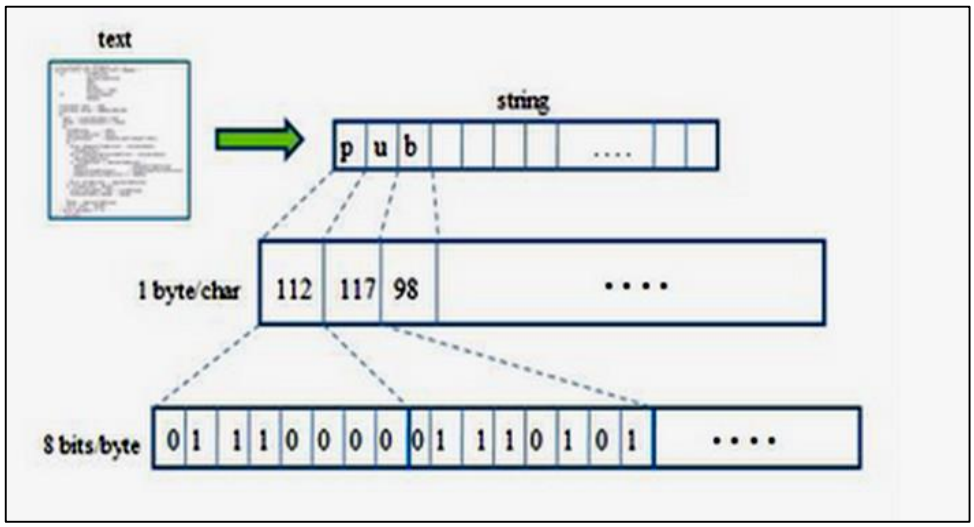
1. Відкрийте файл-контейнер, з якого ви хочете відновити приховане повідомлення;
2. Виділіть наймолодші значущі біти з кожного елемента контейнера (наприклад, пікселя);
3. Складіть біти разом, утворюючи повідомлення в бінарному форматі;
4. Перетворіть бінарне повідомлення в потрібний формат файлу (наприклад, текстовий файл, зображення або аудіофайл).

Важливо зазначити, що процес відтворення може бути успішним тільки в тому випадку, якщо в контейнері було достатньо "вільних" LSB для приховування всього

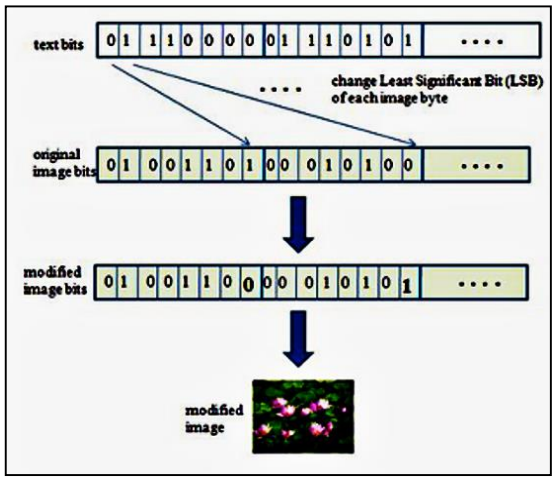
повідомлення. Крім того, приховане повідомлення може бути вразливим до втрати або пошкодження, оскільки вплив на наймолодші значущі біти може вплинути на вихідні дані.

У своїй роботі я буду використовувати наступну структуру, яку описав нижче.

1. Зчитати побайтово дані з файлу, що приховується;



2. Записати кожен байт у двійковому форматі, утворивши послідовність бітів;
3. Зчитати побайтово дані з фалу-контейнеру;
4. Останній один (або кілька) бітів кожного байту замінити на біти секретного повідомлення, візуально можемо дивитись рисунок нижче;



5. Встановити мітку завершення вкраплення прихованого повідомлення;
6. Записати змінену послідовність байтів до стегоконтейнеру.

Алгоритм "заміни наймолодших значущих бітів" є простим у реалізації і надійним методом для приховування файлів у стеганографії. Однак, він також має свої обмеження та викликає певні проблеми, з якими необхідно враховувати при його застосуванні.

Одним з основних обмежень методу "заміни наймолодших значущих бітів" є обмежена ємність для приховування великого обсягу інформації. Оскільки тільки наймолодші значущі біти використовуються для внесення змін, це означає, що обсяг прихованого повідомлення обмежений кількістю пікселів або байтів у контейнері. Великі файли вимагатимуть значних розмірів контейнерів для приховування, що може привести до помітних змін у візуальному вигляді зображення або звуку, що може підвищити підозрілість.

Крім того, метод "заміни наймолодших значущих бітів" також вразливий до різного роду атак та перетворень. Якщо контейнер підданий стисненню або редагуванню після приховування, це може призвести до втрати або пошкодження прихованого повідомлення. Крім того, можлива втрата даних під час передачі чи зберігання файлу, а також зміна формату файлу, можуть вплинути на здатність відновити приховану інформацію.

Для покращення ефективності та стійкості методу "заміни наймолодших значущих бітів" проводяться дослідження та розробка нових підходів та алгоритмів. Наприклад, використання більш значущих бітів, комбінації декількох методів стеганографії, застосування криптографічних методів для забезпечення безпеки інформації. Ці покращення дозволяють збільшити обсяг інформації, яку можна приховати, та забезпечити більшу стійкість до атак та перетворень.

В цілому, метод "заміни наймолодших значущих бітів" є ефективним і широко використовуваним алгоритмом приховування файлів у стеганографії. Його простота та надійність роблять його популярним в різних сферах, включаючи захист даних, комунікації та розвідку. Однак, необхідно бути обережним і враховувати його

обмеження та вразливості, а також використовувати додаткові заходи для забезпечення безпеки інформації.

2.4. Алгоритми видобування файлу із стегоконтейнеру

Алгоритми видобування файлу зі стегоконтейнера включають ряд кроків, які дозволяють отримати приховані дані з обсягового носія. Основні алгоритми для цього процесу включають:

1. Визначення наявності стегоконтейнера: Перший крок полягає в визначенні того, чи наявний стегоконтейнер в досліджуваних даних. Це може включати аналіз заголовків файлів, структурних характеристик або інших ознак, що свідчать про наявність стеганографічної вбудовки.
2. Видобування стегоключа (якщо застосовується): У деяких випадках для видобування прихованих даних може потрібен стегоключ, який використовується для розшифрування стегоконтейнера. Якщо стегоключ доступний, він використовується для розкриття захищеної інформації.
3. Розкриття стегоконтейнера: Наступним кроком є розкриття стегоконтейнера з обсягового носія. Це включає виявлення та виділення прихованої інформації з вхідних даних. Існують різні методи для розкриття стегоконтейнера, які залежать від використовуваного алгоритму стеганографії.
4. Витягнення прихованої інформації: Останній крок полягає в видобуванні самої прихованої інформації з розкритого стегоконтейнера. Це може включати розшифрування даних, якщо використовується шифрування, та повернення відповідного формату чи структури даних.

Залежно від конкретних методів і алгоритмів стеганографії, процес видобування файлу може відрізнятися. Важливо мати на увазі, що алгоритми видобування повинні бути відповідними до використовуваних методів

стеганографії. Нижче наведено кілька популярних алгоритмів видобування файлу зі стегоконтейнера:

LSB Extraction вбудоване повідомлення знаходиться у найменш значущих бітах пікселів зображення. Алгоритм вилучає ці біти і формує приховане повідомлення.

Statistical Analysis: Цей алгоритм базується на аналізі статистичних властивостей стегоконтейнера. Він виявляє відхилення від очікуваного розподілу пікселів чи байтів у зображенні, що може свідчити про наявність прихованої інформації. Алгоритм використовує статистичні методи для виявлення та видобування цих відхилень.

Payload Extraction (Вилучення за допомогою навантаження): Цей алгоритм базується на структурі стегоконтейнера та специфічних ознаках, які використовуються для вбудовування прихованої інформації. Він аналізує структуру файлу, виявляє маркери або особливості, що вказують на наявність прихованої інформації, і видобуває цю інформацію згідно зі специфікаціями стеганографічного методу.

Transform Domain Extraction (Вилучення в області перетворення): Цей алгоритм використовує перетворення, такі як Дискретне Косинусне Перетворення (DCT) або Вейвлет-перетворення, для видобування прихованої інформації зі стегоконтейнера. Він використовує особливості спектрального представлення даних та виявляє зміни, що відбулися в результаті вбудовування повідомлення.

Neural Network-based Extraction (Вилучення на основі нейронних мереж): З використанням розширення застосування штучних нейронних мереж до області стеганографії, розроблені методи, які використовують нейронні мережі для видобування прихованої інформації. Ці методи можуть аналізувати структуру та характеристики стегоконтейнера, щоб визначити наявність та видобути приховане повідомлення.

Важливо зазначити, що алгоритми видобування файлу можуть варіюватися в залежності від використовуваних методів стеганографії, типу стегоконтейнера та особливостей досліджуваних даних. Враховуючи це, розробники стеганалітичних

алгоритмів постійно вдосконалюють методи видобування для забезпечення ефективного виявлення та видобування прихованої інформації.

У своїй роботі я буду використовувати метод LSB, так як цей метод є відносно простим, але ефективність можна суттєво підвищити, якщо створити дворівневий захист з використанням захисту приховуваних даних паролем чи криптографії, зламати який практично неможливо.

Рекомендується використовувати як файли-контейнери для методу LSB такі файли, як скановані документи або фотографії, які були отримані зі сканера чи фотокамери. Ці файли мають природні шуми та відхилення, що ускладнюють виявлення вбудованого повідомлення.

2.5. Метод Least Significant Bit

Метод "заміни наймолодших значущих бітів" (LSB) є одним з найпоширеніших та ефективних методів приховування файлів у стеганографії. Цей метод базується на заміні найменш значущих бітів цифрового контейнера, такого як зображення або звуковий файл, на біти повідомлення, яке ми бажаємо приховати. Застосування методу LSB дозволяє втратити мінімальну кількість інформації та мінімізувати помітність змін у контейнері.

Процес приховування файлів за допомогою методу LSB може бути розбитий на кілька етапів. Спочатку вибирається контейнер, який буде використовуватись для приховування повідомлення. Цим контейнером може бути зображення, звуковий або відео файл. Далі обирається файл, який ми бажаємо приховати, і його перетворюємо у бінарний формат, розбиваючи його на окремі біти.

Наступним кроком є заміна наймолодших значущих бітів контейнера на біти повідомлення. Якщо контейнер має піксельну структуру, то наймолодші значущі біти кожного пікселя замінюються на відповідні біти повідомлення. Цей процес може бути повторений для кожного пікселя чи іншого елемента контейнера, залежно від його структури.

У методі LSB також існує процес відтворення прихованого повідомлення з контейнера. Спочатку відкривається файл-контейнер, з якого ми бажаємо відновити приховане повідомлення. Потім виконується виділення наймолодших значущих бітів з кожного елемента контейнера, наприклад, пікселя. Зібрані біти потім складаються разом, утворюючи повідомлення в бінарному форматі.

Остаточним кроком є перетворення бінарного повідомлення в потрібний формат файлу, наприклад, текстовий файл, зображення або аудіофайл. При цьому важливо враховувати, що успішне відтворення повідомлення можливе лише тоді, коли в контейнері було достатньо "вільних" LSB для приховування всього повідомлення. Крім того, приховане повідомлення може бути вразливим до втрати або пошкодження, оскільки зміна наймолодших значущих бітів може вплинути на вихідні дані.

У моїй роботі я планую використовувати наступну структуру для методу LSB. Спочатку буде здійснено побайтове зчитування даних з файлу, який ми бажаємо приховати. Кожен байт буде перетворений у двійковий формат, утворюючи послідовність бітів. Потім буде здійснено побайтове зчитування даних з фалу-контейнера, а останній один (або кілька) бітів кожного байту будуть замінені на біти секретного повідомлення. Після цього буде встановлена мітка завершення вкраплення прихованого повідомлення, і змінена послідовність байтів буде записана до стегоконтейнера.

На діаграмі рис.2.5. показано наступні кроки:

1. Ми починаємо з контейнера, який є файлом з прихованим повідомленням;
2. Файл відтворюється;
3. Використовується метод LSB (Least Significant Bit) для витягування бітів повідомлення;
4. Витягнуті біти перетворюються на текстове повідомлення;
5. В результаті ми отримуємо приховане повідомлення.

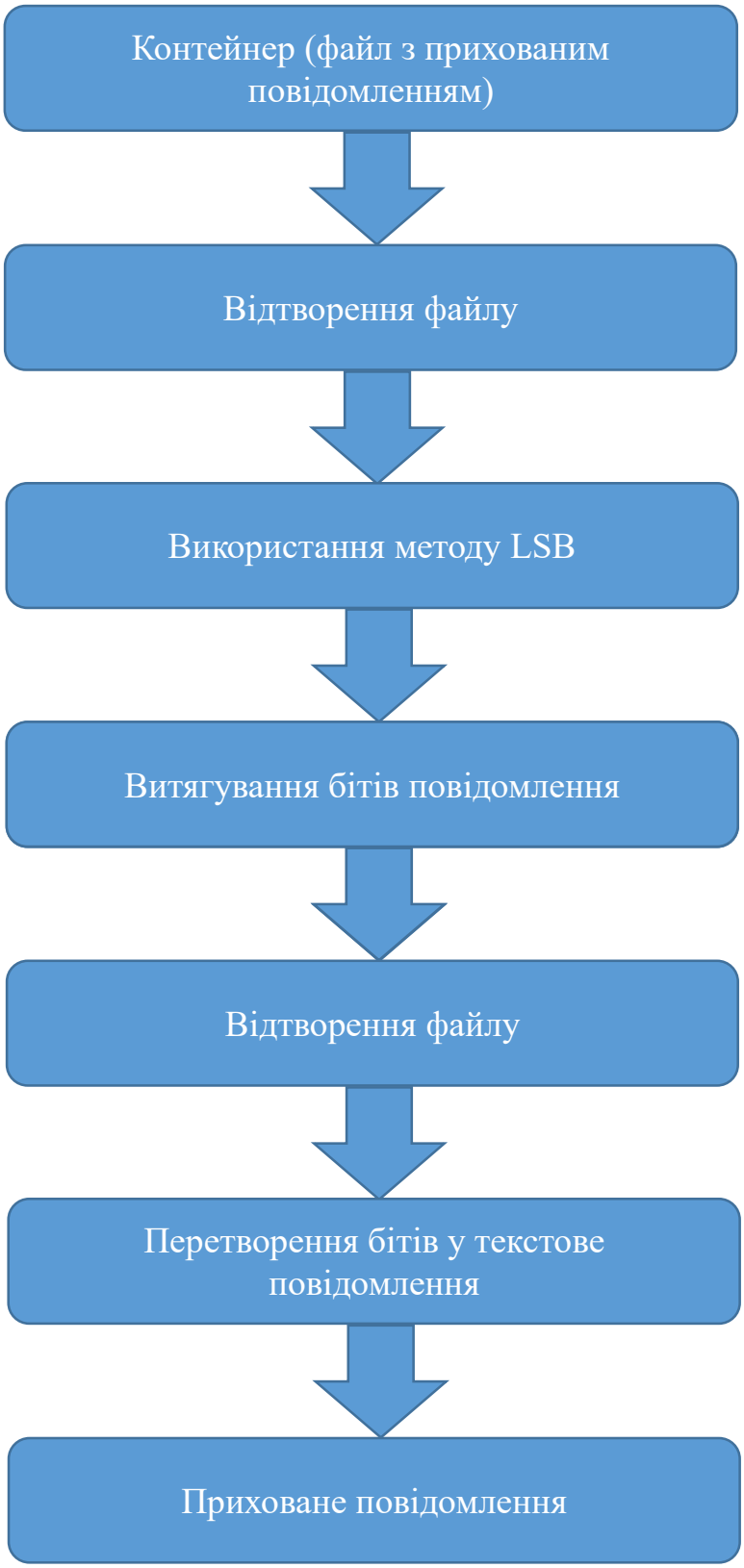


Рис.2.5. Діаграма алгоритму LSB

Метод LSB є потужним і універсальним інструментом для приховування файлів у цифрових контейнерах. Він забезпечує високу стійкість та мінімальну

помітність змін у контейнері. Однак, при використанні методу LSB необхідно враховувати його обмеження та потенційні загрози безпеці, такі як можливість виявлення прихованої інформації або вплив на якість контейнера. Тому, при застосуванні методу LSB, слід добре продумати та врахувати всі аспекти, пов'язані з безпекою та надійністю приховування.

2.6. Метод Statistical Analysis

В рамках дипломної роботи на тему "Застосування прихованої передачі повідомлень на основі цифрової стеганографії" було розглянено різні методи приховування інформації в носіях даних. Один з ключових методів, який ми вивчаємо, - Statistical Analysis (статистичний аналіз). Цей метод виявляється дуже важливим у сфері цифрової стеганографії, оскільки він дозволяє виявити приховані повідомлення шляхом аналізу статистичних властивостей носіїв даних, таких як зображення, звукові файли та відео.

Застосування статистичного аналізу в цифровій стеганографії є важливою галуззю досліджень, оскільки воно дозволяє виявляти та аналізувати приховану інформацію з високою точністю. Цей метод може бути корисним для захисту конфіденційної інформації та запобігання незаконному використанню даних.

Статистичний аналіз - це важливий інструмент, який допомагає нам розуміти та інтерпретувати дані. Він включає в себе різноманітні методи, які дозволяють нам описувати, оцінювати та порівнювати дані.

Статистичний аналіз можна розглядати як процес, що включає збір, організацію, аналіз, інтерпретацію та представлення даних. Він допомагає нам виявляти шаблони, встановлювати зв'язки між змінними та робити висновки на основі наших спостережень.

Основні етапи статистичного аналізу включають:

1. Збір даних: Це перший крок у процесі статистичного аналізу. Дані можуть бути зібрані з різних джерел, таких як опитування, експерименти, спостереження або вже існуючі набори даних.

2. Організація даних: Після збору даних їх потрібно організувати таким чином, щоб їх було легко аналізувати. Це може включати в себе створення таблиць, графіків або діаграм.
3. Аналіз даних: На цьому етапі використовуються статистичні методи для вивчення даних. Це може включати в себе описову статистику, інференційну статистику або прогнозування.
4. Інтерпретація даних: Після аналізу даних ми можемо робити висновки або висновки на основі наших результатів. Це може включати в себе визначення значимості результатів, виявлення зв'язків між змінними або визначення тенденцій.
5. Представлення даних: Останній етап статистичного аналізу - це представлення наших результатів . Це може включати в себе створення звітів, презентацій або візуалізацій даних.

Діаграма, яка ілюструє цей процес зображена на рис.2.6.:

Рис.2.6. Діаграма Statistical Analysis

Статистичний аналіз відіграє важливу роль в багатьох галузях, включаючи науку, технології, бізнес, охорону здоров'я та уряд. Він допомагає нам робити обґрунтовані рішення, прогнозувати майбутні тенденції та виявляти проблеми.

Важливо зазначити, що статистичний аналіз - це не просто процес обробки чисел. Він включає в себе критичне мислення, інтерпретацію даних та розуміння контексту, в якому ці дані використовуються.

Під час проведення статистичного аналізу для виявлення прихованих повідомлень у носіях даних, ми звертаємо увагу на різні статистичні характеристики та їх зміни. Наприклад, в зображеннях, ми досліджуємо розподіл яскравості або кольору пікселів. За нормальних умов, нерозмічені зображення будуть мати певний розподіл, який можна описати статистичною моделлю, наприклад, нормальним розподілом. Але якщо у зображенні присутні приховані повідомлення, то це може призвести до змін у розподілі яскравості або кольору пікселів. Використовуючи статистичні тести, такі як тест Колмогорова-Смірнова або тест хі-квадрат, ми можемо виявити ці зміни і підтвердити наявність прихованого повідомлення.

Крім аналізу розподілу пікселів, ми також досліджуємо кореляційні залежності між пікселями у зображеннях. Приховані повідомлення можуть вплинути на ці кореляції, і ми використовуємо методи, такі як кореляційна матриця або автоковаріаційна функція, щоб виявити ці зміни. Якщо існують несподівані кореляції або некореляції між пікселями, це може свідчити про наявність прихованого повідомлення.

У випадку аудіофайлів, статистичний аналіз зазвичай проводиться на основі спектральних характеристик звукових сигналів. Ми можемо використовувати методи, такі як спектральний аналіз чи аналіз амплітуди, для виявлення змін, які можуть виникнути при приховані повідомлення в аудіофайлі. Зміни у спектральних характеристиках, наприклад, несподівані піки або затримки, можуть свідчити про наявність прихованої інформації.

Відеофайли також можуть бути аналізовані з використанням статистичних методів, які дозволяють виявляти зміни у статистичних властивостях кадрів. Наприклад, ми можемо досліджувати яскравість, кольоровий тон або текстурні

характеристики кадрів і шукати несподівані зміни, які можуть виникнути при наявності прихованої інформації.

Важливою складовою статистичного аналізу є вибір відповідних статистичних тестів та методів, які забезпечать точні та надійні результати. При проведенні статистичного аналізу слід враховувати можливі фактори, які можуть вплинути на його результати, такі як шум, стиснення даних або різниця в якості носіїв.

Використання статистичного аналізу дозволяє нам краще розуміти світ навколо нас. Він допомагає нам виявляти шаблони, робити висновки та робити прогнози на основі даних. Це важливий інструмент, який допомагає нам робити обґрунтовані рішення в різних областях життя.

2.7. Метод Payload Extraction

Метод Payload Extraction є одним з ключових методів стеганографії, який використовується для видобування прихованої інформації з цифрових медіа-контейнерів. Цей метод дозволяє ефективно витягувати приховане повідомлення, яке було вбудовано в контейнер за допомогою різних методів, наприклад, методу LSB (Least Significant Bit) або методу частотного аналізу.

Для успішного витягування повідомлення за методом Payload Extraction необхідно виконати декілька кроків. Перш за все, необхідно відкрити цифровий медіа-контейнер, з якого ми хочемо витягнути приховану інформацію. Далі, застосовуючи відповідний алгоритм, ми виділяємо приховане повідомлення з контейнера.

Один із підходів до видобування повідомлення за методом Payload Extraction полягає у виявленні особливих властивостей, які використовуються при вбудовуванні прихованої інформації. Наприклад, в методі LSB, кожен біт прихованого повідомлення заміщає найменш значущий біт пікселя в цифровому зображенні. Таким чином, в процесі видобування повідомлення ми аналізуємо структуру контейнера, виявляємо модифіковані біти і відновлюємо приховане повідомлення.

Існує також інший підхід до видобування повідомлення - метод частотного аналізу. Цей метод базується на виявленні статистичних аномалій або нерегулярностей у цифровому контейнері, що можуть вказувати на наявність прихованої інформації. Аналізуючи розподіл частот або інші статистичні показники контейнера, можна виявити вбудоване повідомлення і витягнути його.

Порівнюючи методи Payload Extraction, важливо враховувати їх переваги та недоліки. Наприклад, метод LSB є простим у реалізації та має високу швидкодію, але приховане повідомлення може бути вразливе до атак, що спрямовані на виявлення. З іншого боку, метод частотного аналізу є більш стійким до атак, проте вимагає більшої обчислювальної потужності.

2.8. Метод Transform Domain Extraction

Метод Transform Domain Extraction є ще одним ефективним методом стеганографії, який використовується для видобування прихованої інформації з цифрових медіа-контейнерів. Цей метод базується на використанні перетворень домену, таких як дискретне косинусне перетворення (DCT) або хвильове перетворення, для розкриття прихованої інформації.

Для використання методу Transform Domain Extraction необхідно виконати кілька кроків. Початково, цифровий медіа-контейнер, який містить приховану інформацію, піддається певному перетворенню домену, наприклад, DCT. Після цього відбувається аналіз коефіцієнтів перетворення, щоб виявити наявність прихованої інформації.

Одним з ключових аспектів методу Transform Domain Extraction є використання порогового значення для визначення, які коефіцієнти перетворення відповідають прихованій інформації. Зазвичай, найменш значущі коефіцієнти вважаються менш важливими для зображення, тому вони використовуються для вбудовування прихованої інформації. В процесі видобування прихованого повідомлення використовуються порігові значення, які дозволяють виявити та витягнути приховану інформацію з коефіцієнтів перетворення.

2.9. Метод Neural Network-based Extraction

Метод "Neural Network-based Extraction" є одним зі способів видобування прихованої інформації з медіа-контейнерів за допомогою нейромережевих моделей. Цей метод базується на використанні глибоких нейронних мереж, що дозволяють здійснювати складні аналізи зображень, звуків або відео з метою виявлення та витягування прихованих даних.

В сучасному світі, де цифрова інформація стає все більш цінною та конфіденційною, методи стеганографії, як Neural Network-based Extraction, знаходять широке застосування. Вони дозволяють приховувати повідомлення в різних типах медіа-даних, таких як зображення, звук і відео, з метою забезпечення конфіденційності та безпеки передачі інформації.

Метод Neural Network-based Extraction використовує глибокі нейронні мережі, які навчаються розпізнавати та видобувати приховані повідомлення з медіа-контейнерів. Цей процес передбачає два основних етапи: тренування моделі і застосування моделі для видобування.

У першому етапі, для тренування нейромережевої моделі потрібно мати набір тренувальних даних, який складається з прихованих повідомлень та відповідних медіа-контейнерів, в яких ці повідомлення були приховані. Тренувальні дані мають різні характеристики та параметри, що відображають різноманітні сценарії стеганографії. Ці дані використовуються для тренування моделі з метою навчитися розпізнавати та витягувати приховані повідомлення з інших медіа-контейнерів.

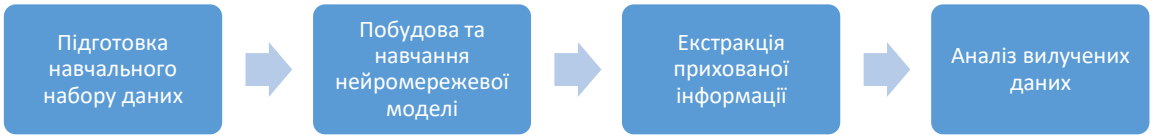
Другий етап полягає в застосуванні навченої моделі до нових медіа-контейнерів для видобування прихованих повідомлень. Цей процес включає обробку медіа-контейнера за допомогою нейромережевої моделі, яка аналізує дані, виявляє наявність прихованих повідомлень і витягує їх. Результатом є відновлене приховане повідомлення, яке може бути далі використане для подальшого аналізу або використано у відповідних цілях.

Однією з головних переваг методу Neural Network-based Extraction є його здатність до роботи з різними типами медіа-даних та розпізнавання прихованих

повідомлень незалежно від їх формату. Це робить метод універсальним та ефективним, оскільки він може бути застосований до широкого спектру стеганографічних сценаріїв і задач.

Використання нейромережевих моделей для видобування прихованих повідомлень відкриває широкі перспективи для подальшого розвитку методів стеганографії. Навчання глибоких нейронних мереж на великих обсягах тренувальних даних може покращити точність та швидкодію видобування, а також розширити можливості застосування методу.

Для кращого уявлення процесу Neural Network-based Extraction та його етапів, нижче наведена діаграма:



Підсумовуючи, метод Neural Network-based Extraction є потужним інструментом для видобування прихованої інформації з медіа-контейнерів. Використання глибоких нейронних мереж дозволяє досягти високої точності і швидкодії видобування, що робить цей метод привабливим для дослідження та реалізації у практичних застосуваннях стеганографії. Такий підхід до видобування прихованих повідомлень відкриває нові можливості для забезпечення безпеки та конфіденційності передачі інформації у різних сферах, таких як кібербезпека, комунікації та цифрові медіа.

2.10. Висновок до розділу 2

У даному розділі було розглянуто алгоритм приховування повідомлень методом заміни найменших значущих бітів у цифровій стеганографії. Проаналізувавши принципи роботи методів цифрової стеганографії, ми перейшли до вивчення структури графічних файлів формату BMP, яка є основою для приховування повідомлень у зображеннях.

В рамках розділу були описані алгоритми приховування файлів методом "заміни наймолодших значущих бітів" та їх відтворення. Ці алгоритми дозволяють ефективно вбудовувати інформацію у зображення, замінюючи найменш значущі біти пікселів. Також були представлені алгоритми видобування файлів із стегоконтейнерів, що дозволяють витягти приховану інформацію зі зображень.

Вивчивши методи стеганалізу, такі як метод Least Significant Bit, метод Statistical Analysis, метод Payload Extraction, метод Transform Domain Extraction та метод Neural Network-based Extraction, ми отримали уявлення про різні підходи до виявлення та видобування прихованої інформації зі зображень.

Результати досліджень у цьому розділі підтверджують ефективність та застосовність методу заміни найменших значущих бітів для цифрової стеганографії. Застосування цього методу дозволяє надійно та безпечно приховувати повідомлення у зображеннях, забезпечуючи високий рівень захисту і надійності інформації.

У подальших розділах нашої пояснювальної записки ми будемо розглядати застосування цифрової стеганографії з використанням мови програмування Python, побудову стегоконтейнерів за допомогою зображень у форматі BMP, а також реалізацію алгоритмів приховування та видобування файлів зі стегоконтейнерів у мові Python.

Отже, на основі проведених досліджень можна зробити висновок, що метод заміни найменших значущих бітів є ефективним та практичним підходом до цифрової стеганографії. Його застосування в сучасному інформаційному суспільстві відкриває широкі можливості для безпечного обміну конфіденційною інформацією та забезпечує високий рівень захисту даних. Дані алгоритми та методи можуть бути

успішно реалізовані у мові програмування Python, що сприяє їх широкому застосуванню та розповсюдженню серед різних галузей.

РОЗДІЛ 3.
ЗАСТОСУВАННЯ ЦИФРОВОЇ СТЕГАНОГРАФІЇ З ВИКОРИСТАННЯМ МОВИ ПРОГРАМУВАННЯ PYTHON

3.1. Огляд мови програмування Python та її можливості

Python є однією з найпопулярніших мов програмування у світі, завдяки своїй простоті, ефективності та широкому спектру можливостей. У цьому підрозділі ми розглянемо деякі ключові особливості мови Python, які зробили її популярною серед розробників ідеальним інструментом для реалізації алгоритмів стеганографії.

Однією з найбільших переваг Python є його простий синтаксис, який дозволяє розробникам швидко створювати та виконувати програми без зайвих зусиль. Це особливо важливо в контексті стеганографії, де потрібно працювати з великою кількістю даних та виконувати операції з пікселями зображень чи аудіофайлів.

Можливості мови Python: Python підтримує різноманітні функції та функціональності, які роблять його потужним інструментом для стеганографії. Наприклад, Python має вбудовані функції для роботи з рядками, списками, словниками та іншими типами даних, що спрощує обробку та маніпуляцію даними в стеганографії.

Python також відомий своїми багатими бібліотеками та модулями, які спрощують реалізацію складних алгоритмів стеганографії. Наприклад, бібліотека PIL (Python Imaging Library) дозволяє працювати з зображеннями та виконувати операції з пікселями. Це дозволяє розробникам створювати стегоконтейнери з використанням зображень у форматі BMP та змінювати значення пікселів для приховування додаткової інформації.

Простота використання: Python є інтуїтивно зрозумілою мовою програмування, яка дозволяє розробникам швидко розробляти та відлагоджувати код. Він має зрозумілі конструкції та синтаксис, що сприяє швидкій розробці програм та полегшує їх збереження та обслуговування.

Швидкодія: Python є інтерпретованою мовою, але завдяки оптимізаціям та використанню сторонніх бібліотек, таких як NumPy, можливо досягти високої швидкодії обробки даних. Це особливо важливо при роботі з великими обсягами даних у стеганографії.

Кросплатформеність: Python підтримується на різних операційних системах, включаючи Windows, macOS та Linux, що дозволяє розробляти стеганографічні програми, які працюють на різних платформах без змін коду.

Розширюваність: Python має можливість інтеграції з іншими мовами програмування, такими як C/C++, що дозволяє реалізовувати високопродуктивні алгоритми стеганографії, які вимагають більшої швидкодії.

Застосування мови програмування Python у цифровій стеганографії виявляється дуже ефективним і зручним. Python має потужну та просту синтаксичну структуру, що дозволяє легко розробляти стеганографічні програми та виконувати їх без проблем. Наявність розширених бібліотек, таких як PIL і NumPy, спрощує роботу з зображеннями та числовими даними, що є важливими в контексті стеганографії.

Python також надає можливості роботи з файлами, мережами та шифруванням, що робить його універсальним інструментом для реалізації різних алгоритмів стеганографії. Існують спеціалізовані бібліотеки, які дозволяють швидко імплементувати різні методи стеганографії, такі як метод найменших значущих бітів та нейронні мережі.

Загалом, Python є потужним та гнучким інструментом для стеганографії, і його використання дозволяє швидко розробляти та реалізовувати різноманітні алгоритми стеганографії з високою ефективністю. Використання Python у поєднанні зі спеціалізованими бібліотеками та пакетами робить його ідеальним вибором для стеганографічних досліджень та розробок.

3.2. Побудова стегоконтейнерів за допомогою зображень у форматі bmp

У цьому підрозділі ми розглянемо побудову стегоконтейнерів за допомогою зображень у форматі BMP з використанням мови програмування Python. Стегоконтейнери є носіями, в які вбудовується прихована інформація за допомогою стеганографічних алгоритмів. Формат BMP (Bitmap) є одним з найпоширеніших форматів зображень, який зберігає графічні дані у вигляді матриці пікселів.

Ми розглянемо кроки, необхідні для побудови стегоконтейнерів, використовуючи мову програмування Python.

1. Завантаження зображення BMP: Ми починаємо з завантаження зображення BMP, яке буде використовуватися як стегоконтейнер. Наприклад, ми можемо використовувати бібліотеку PIL (Python Imaging Library) для цієї операції. Ось приклад коду на рис.3.1:

```
from PIL import Image

# Завантаження зображення BMP
image = Image.open('image.bmp')
```

Рис.3.1. Завантаження зображення з використанням бібліотеки PIL

2. Зчитування додаткової інформації: Потім ми вибираємо файл або текст, який ми хочемо приховати у стегоконтейнері. Ми можемо зчитувати байти або символи з файлу та перетворювати їх на бітовий формат. Ось приклад коду для зчитування тексту на рис.3.2:

```
# Зчитування тексту, який буде приховуватися
message = "Приховане повідомлення"
bits = ''.join(format(ord(c), '08b') for c in message)
```

Рис.3.2. Приклад коду для зчитування тексту

3. Побудова стегоконтейнера: Далі, ми проходимося по кожному пікселю зображення та замінюємо його найменш значущі біти на біти з додаткової інформації. Наприклад, якщо ми хочемо замінити 2 найменш значущих біти червоного каналу, ми можемо використовувати операцію побітового зсуву та побітового I для заміни бітів. Ось приклад коду на рис.3.3:

```
# Побудова стегоконтейнера
for i in range(len(bits)):
    pixel = image.getpixel((i % image.width, i // image.width))
    red = pixel[0]
    new_red = (red & 0xFC) | int(bits[i:i+2], 2)
    new_pixel = (new_red, pixel[1], pixel[2])
    image.putpixel((i % image.width, i // image.width), new_pixel)
```

Рис.3.3. Побудова стегоконтейнера

4. Збереження стегоконтейнера: Нарешті, ми зберігаємо отримане стегоконтейнерне зображення. Ось приклад коду на рис.3.4:

```
# Збереження стегоконтейнера
image.save('stego_image.bmp')
```

Рис.3.4. Збереження стегоконтейнера

Цей процес побудови стегоконтейнера з використанням зображень у форматі BMP дозволяє нам приховувати додаткову інформацію у зображеннях. У наступному пункті ми розглянемо алгоритми для видобування прихованої інформації зі стегоконтейнерів у мові Python.

3.3. Реалізація алгоритмів приховування файлів методом "заміни найменших значущих бітів" у мові Python

У цьому пункті ми розглянемо реалізацію алгоритмів приховування файлів методом "заміни найменших значущих бітів" у мові програмування Python. Цей метод дозволяє приховати файл в стегоконтейнері, замінюючи його біти на найменш значущі біти пікселів зображення. Давайте розглянемо кроки реалізації цього алгоритму:

1. Завантаження стегоконтейнера: Починаємо з завантаження стегоконтейнерного зображення, в якому будемо приховувати файл. Зображення повинно бути у форматі BMP. Ось приклад коду для завантаження стегоконтейнера на рис.3.5:

```
from PIL import Image

# Завантаження стегоконтейнера
stego_image = Image.open('stego_image.bmp')
```

Рис.3.5. Збереження стегоконтейнера

2. Завантаження файлу для приховування: Вибираємо файл, який ми хочемо приховати у стегоконтейнері. Наприклад, це може бути текстовий файл, аудіо-файл або будь-який інший тип файлу. Ось приклад коду для завантаження файлу на рис.3.6:

```
# Завантаження файлу для приховування
file_path = 'secret_file.txt'
with open(file_path, 'rb') as file:
    file_data = file.read()
```

Рис.3.6. Завантаження файлу

3. Приховування файлу в стегоконтейнері: Проходимося по кожному байту файлу та замінюємо його найменш значущі біти пікселів стегоконтейнера. Для цього ми використовуємо операцію побітового зсуву та побітового І для заміни бітів. Ось приклад коду для приховування файлу на рис.3.7.:

```
# Приховування файлу в стегоконтейнері
file_index = 0
for i in range(stego_image.width * stego_image.height):
    pixel = stego_image.getpixel((i % stego_image.width, i // stego_image.width))
    red = pixel[0]
    new_red = (red & 0xFE) | ((file_data[file_index] >> 7) & 0x01)
    new_pixel = (new_red, pixel[1], pixel[2])
    stego_image.putpixel((i % stego_image.width, i // stego_image.width), new_pixel)

    file_data[file_index] = (file_data[file_index] << 1) & 0xFF
    file_index = (file_index + 1) % len(file_data)
```

Рис.3.7. Приховування файлу

4. Збереження стегоконтейнера: Нарешті, ми зберігаємо отриманий стегоконтейнерне зображення з прихованою інформацією. Ось приклад коду для збереження стегоконтейнера на рис.3.8.:

```
# Збереження стегоконтейнера
stego_image.save('stego_image_with_file.bmp')
```

Рис.3.8. Збереження стегоконтейнера

Цей алгоритм реалізує метод "заміни найменших значущих бітів" для приховування файлів у стегоконтейнері за допомогою зображень у форматі BMP. У наступному пункті ми розглянемо алгоритми для видобування прихованої інформації зі стегоконтейнерів у мові Python.

3.4. Реалізація алгоритмів видобування файлів із стегоконтейнерів у мові Python

У цьому пункті ми розглянемо реалізацію алгоритмів для видобування файлів із стегоконтейнерів у мові програмування Python. Після приховування файлу в стегоконтейнері, ми можемо використовувати ці алгоритми для видобування прихованої інформації. Розглянемо кроки реалізації цих алгоритмів:

1. Завантаження стегоконтейнера з прихованою інформацією: Починаємо з завантаження стегоконтейнера, який містить приховану інформацію. Використовуємо зображення у форматі BMP, яке було створене після приховування файлу. Ось приклад коду для завантаження стегоконтейнера на рис.3.9.:

```
from PIL import Image

# Завантаження стегоконтейнера з прихованою інформацією
stego_image = Image.open('stego_image_with_file.bmp')
```

Рис.3.9. Завантаження стегоконтейнера

2. Видобування файлу зі стегоконтейнера: Проходимося по кожному пікселю стегоконтейнера та витягуємо найменш значущі біти, які містять приховану інформацію. Зібрані біти формують прихований файл. Ось приклад коду для видобування файлу на рис.3.10.:

```
# Видобування файлу зі стегоконтейнера
extracted_file_data = bytearray()
for i in range(stego_image.width * stego_image.height):
    pixel = stego_image.getpixel((i % stego_image.width, i // stego_image.width))
    red = pixel[0]
    extracted_file_data.append(red & 0x01)
```

Рис.3.10. Видобування файлу

У цьому коді ми зчитуємо пікселі стегоконтейнера та витягуємо найменш значущий біт червоного компонента кольору. Цей біт додається до видобутого файлу. Процес повторюється для кожного пікселя стегоконтейнера, доки не буде видобуто весь файл.

3. Збереження видобутого файлу: Нарешті, ми зберігаємо видобутий файл на локальний диск. Ось приклад коду для збереження файлу на рис.3.11:

```
# Збереження видобутого файлу
with open('extracted_file.txt', 'wb') as file:
    file.write(extracted_file_data)
```

Рис.3.11. Збереження файлу

У цьому коді ми використовуємо відкритий режим файлу 'wb' для збереження видобутого файлу у бінарному форматі.

Ці алгоритми реалізують процес видобування файлів із стегоконтейнерів, що були створені за допомогою методу "заміни найменших значущих бітів" у мові програмування Python. У наступному пункті ми розглянемо використання методу Least Significant Bit для реалізації стеганографії у Python.

3.5. Використання методу Least Significant Bit у мові Python

У цьому пункті ми розглянемо використання методу Least Significant Bit для приховування та видобування інформації у зображеннях у форматі BMP з використанням мови програмування Python на рис.3.12.

```python
from PIL import Image

def hide_data_in_image(image_path, data):
    image = Image.open(image_path)
    modified_image = image.copy()
    binary_data = ''.join(format(byte, '08b') for byte in data)

    data_index = 0

    for i in range(image.width):
        for j in range(image.height):
            pixel = image.getpixel((i, j))
            red, green, blue = pixel

            if data_index < len(binary_data):
                new_red = (red & 0xFE) | int(binary_data[data_index])
                data_index += 1
            else:
                new_red = red

            modified_image.putpixel((i, j), (new_red, green, blue))

    return modified_image

def extract_data_from_image(image_path):
    modified_image = Image.open(image_path)
    extracted_data = bytearray()

    for i in range(modified_image.width):
        for j in range(modified_image.height):
            pixel = modified_image.getpixel((i, j))
            red = pixel[0]
            extracted_data.append(red & 0x01)

    extracted_text = bytes(int(''.join(map(str, extracted_data[i:i+8])), 2) for i in range(0, len(extracted_data), 8))
    return extracted_text
```

Рис.3.12. Приклад коду

У цьому коді ми визначили дві функції: hide_data_in_image, яка приховує дані у зображенні, та extract_data_from_image, яка видобуває приховані дані з зображення. Функція hide_data_in_image приймає шлях до зображення (image_path) та дані, які потрібно приховати (data). Вона завантажує зображення, створює його копію та замінює найменш значущий біт червоного каналу кожного пікселя на біт інформації. Функція повертає модифіковане зображення.

Функція extract_data_from_image приймає шлях до модифікованого зображення (image_path) та видобуває приховані дані з нього. Вона проходиться по кожному пікселю модифікованого зображення та видобуває найменш значущий біт червоного каналу. Потім вона відновлює оригінальну інформацію з отриманої послідовності бітів і повертає її у вигляді байтового рядка.

Цей код демонструє простий приклад використання методу LSB у мові програмування Python для приховування та видобування інформації у зображеннях.

3.6. Реалізація та демонстрація застосунку на основі цифрової стеганографії.

У рамках даної роботи було розроблено інноваційний додаток під назвою Stereo/Destereo, який використовує передові технології цифрової стеганографії для надійного та безпечного приховування та вилучення інформації зображень. Цей додаток надає унікальну можливість користувачам закодовувати конфіденційні дані у зображення формату BMP, а також ефективно декодувати ці дані з закодованих зображень, забезпечуючи максимальну захищеність і надійність інформації.

Основним методом, використаним у додатку Stereo/Destereo, є метод найменш значущого біта (LSB). Цей метод базується на теорії, що найменш значущі біти кольорових компонентів пікселів зображення мають меншу вагу в сприйнятті людським око, тому їх можна використати для приховування інформації без помітного впливу на якість зображення. Додаток Stereo/Destereo використовує цей метод для вкраплення текстових даних у найменш значущі біти пікселів обраного зображення.

Застосунок має інтуїтивний та зручний інтерфейс, що дозволяє користувачам легко вибирати BMP-зображення та текстові файли та виконувати операції кодування та декодування з мінімальними зусиллями. Перед початком процесу кодування, користувач може вибрати бажане зображення та текстовий файл для приховування інформації, використовуючи відповідні кнопки у першій половині вікна. Після успішного закодування, нове зображення з прихованою інформацією відображається у другій половині вікна разом з відповідним підписом. Користувач може зберегти закодоване зображення для подальшого використання або поділитися ним з іншими. Основне вікно нашого застосунку представлено на рис.3.13., забезпечуючи зручну та ефективну роботу з цифровою стеганографією.

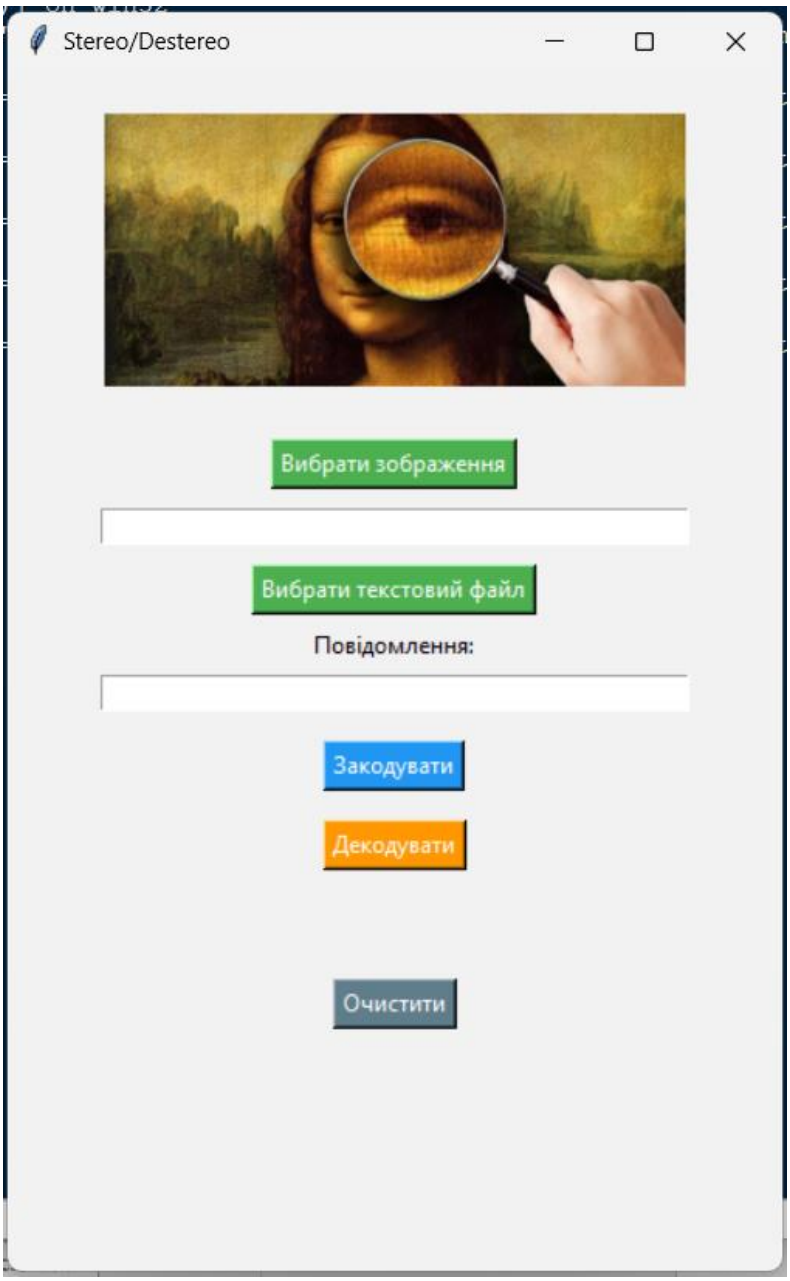


Рис.3.13. Основне вікно застосунку

Для реалізації додатку була використана мова програмування Python та ряд бібліотек, зокрема Tkinter для створення графічного інтерфейсу користувача (GUI), PIL для роботи з зображеннями та стеганографічною бібліотекою stegano.

Реалізація додатку Stereo/Destereo базується на використанні сторонньої бібліотеки Stegano, яка надає потужні функції стеганографії для роботи з зображеннями. Бібліотека Stegano дозволяє легко виконувати операції кодування та декодування зображень, надаючи широкі можливості та гнучкість при роботі з

стеганографією. Вона підтримує різні формати зображень, включаючи BMP, який використовується в даному додатку.

Демонстрація застосунку Stereo/Destereo показала його високу ефективність та потужність у використанні цифрової стеганографії для захисту та передачі конфіденційної інформації. Застосування стеганографії стає все більш актуальним у сучасному світі, де захист конфіденційної інформації має вирішальне значення. Додаток Stereo/Destereo дозволяє користувачам легко та безпечно виконувати ці задачі, забезпечуючи надійну та ефективну стеганографічну обробку зображень.

Результатом реалізації цього додатку є простий та зручний інструмент для стеганографії, який може бути використаний для захисту конфіденційної інформації або для створення цікавих ефектів зображень. Додаток Stereo/Destereo демонструє потужність та гнучкість цифрової стеганографії, а його простий інтерфейс робить його доступним для широкого кола користувачів.

Наш застосунок на основі цифрової стеганографії надає користувачам такий функціонал:

- Вибір зображення;
- Вибір текстового файлу;
- Закодування повідомлення;
- Декодування повідомлення;
- Збереження декодованого повідомлення;
- Очищення.

Вибір зображення: Користувач може вибрати BMP-зображення з файлової системи за допомогою кнопки "Вибір зображення". При виборі зображення відображається у частині вікно, що відображається у наступних рис.3.14. та рис.3.15.

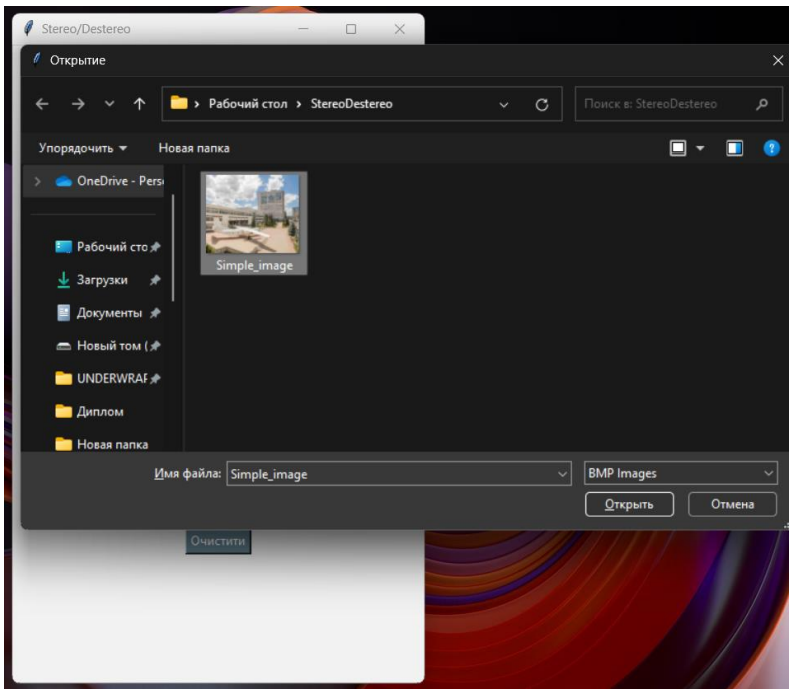


Рис.3.14. Вибір зображення

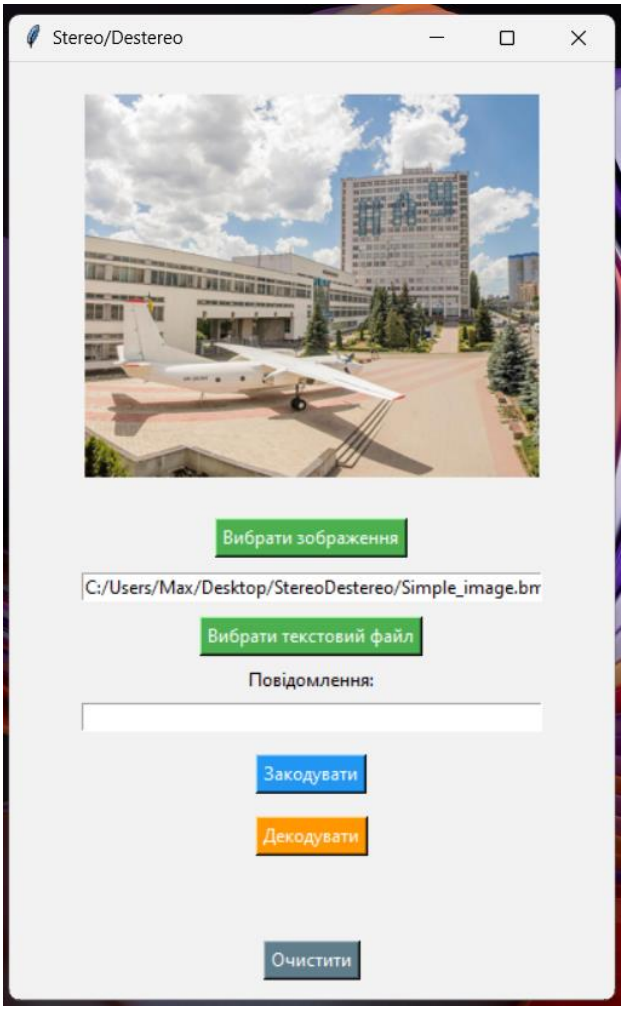


Рис.3.15. Демонстрація шляху до зображення

Вибір текстового файлу: Користувач може вибрати текстовий файл, з якого буде взята інформація для кодування, за допомогою кнопки "Вибір файлу .txt". При виборі файлу з текстом текст чатков відображається у відповідні частині вікна, що зображено у наступних рис.3.16. та рис.3.17.

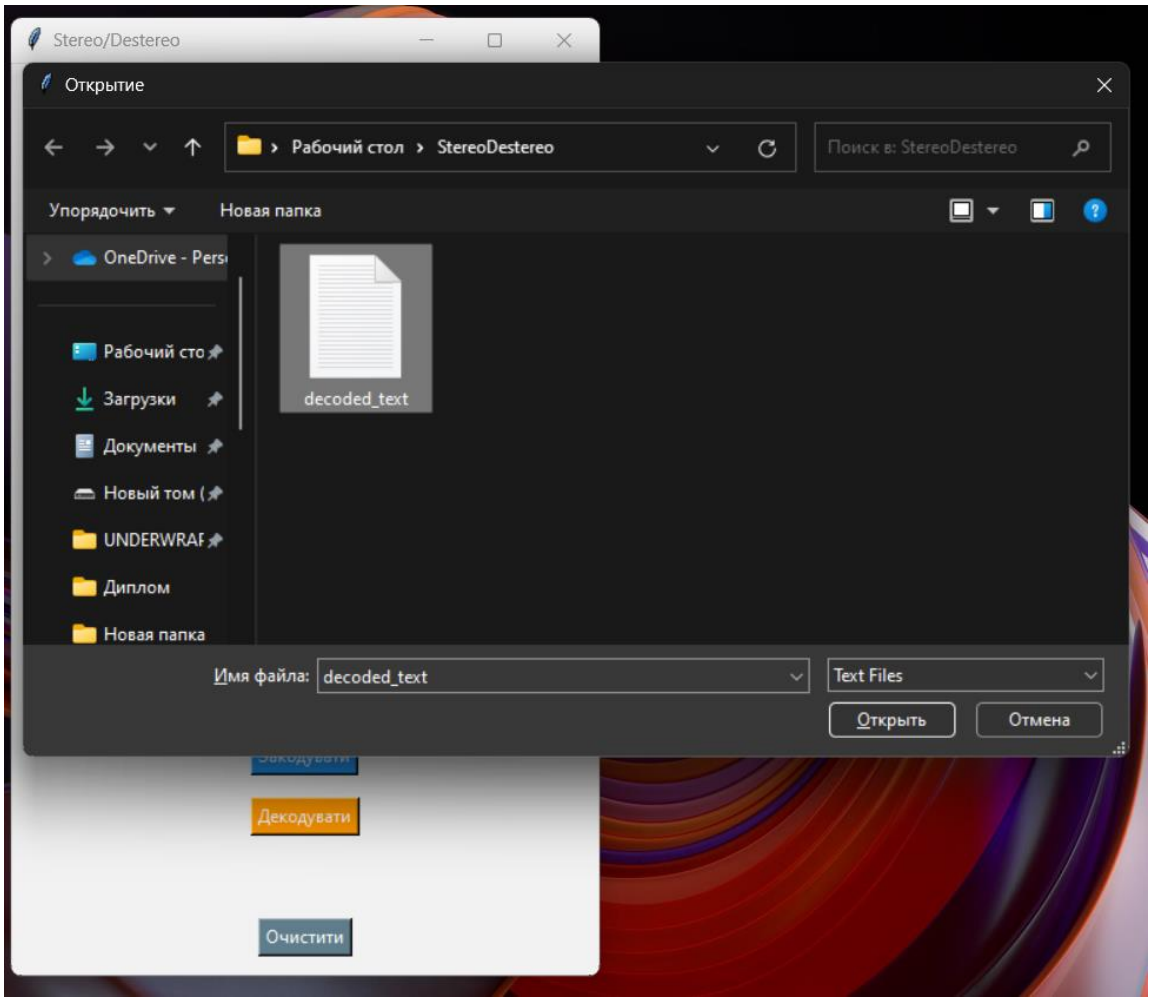


Рис.3.16. Вибір текстового файлу

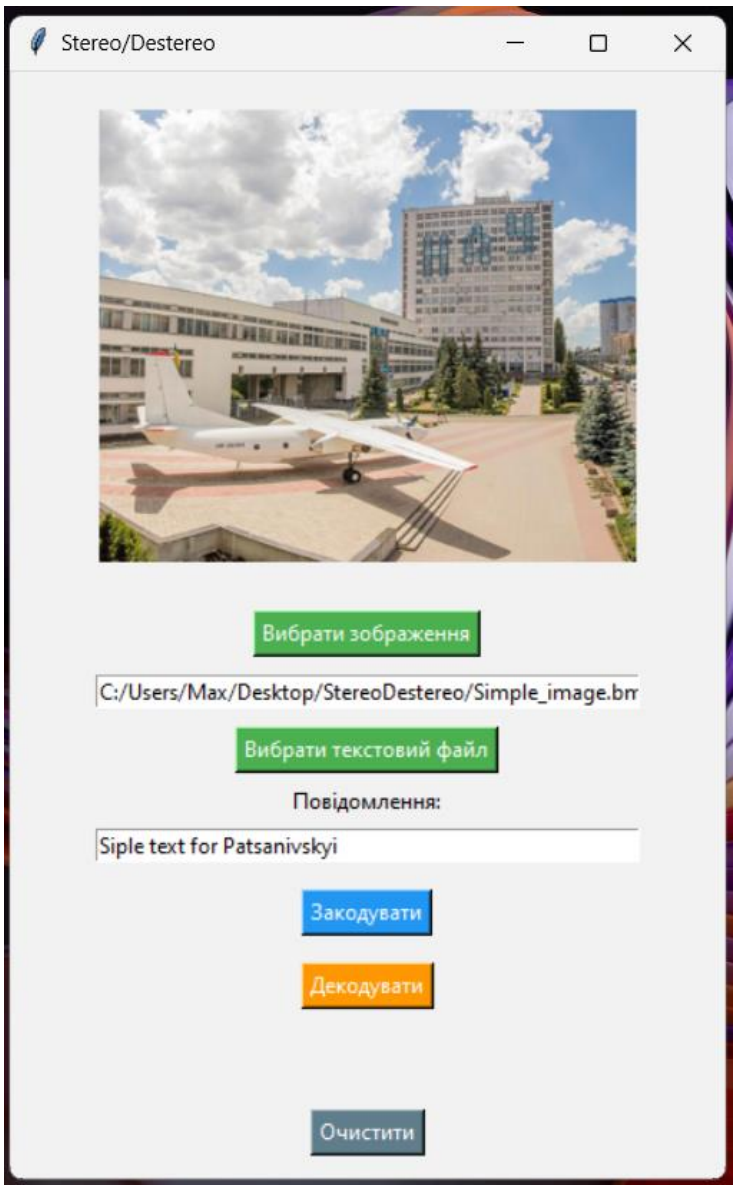


Рис.3.17. Відображення тексту

Закодування повідомлення: При натисканні кнопки "Закодувати", обране зображення та текстовий файл піддаються обробці. Текстові дані зчитуються з файлу, а потім закодовуються у вибране зображення за допомогою методу LSB (Least Significant Bit). Результатом є нове зображення з прихованою інформацією, яке відображається у вікні разом з відповідним підписом, показано на рис.3.18. Зображення зберігається під назвою «Початкова назва зображення_stereo» та поміщається в папку початкового зображення це відобразив на рис.3.19

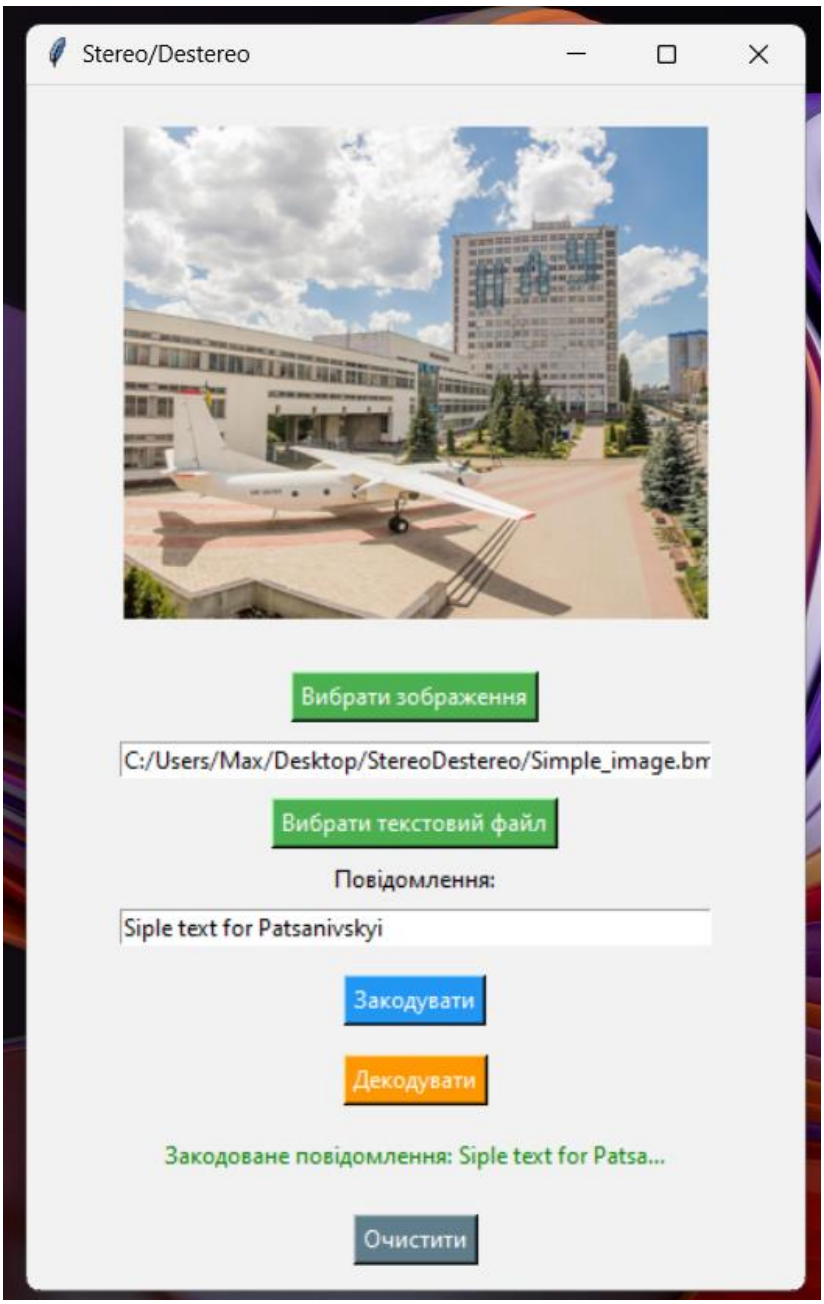


Рис.3.18. Повідомлення про успішне виконання

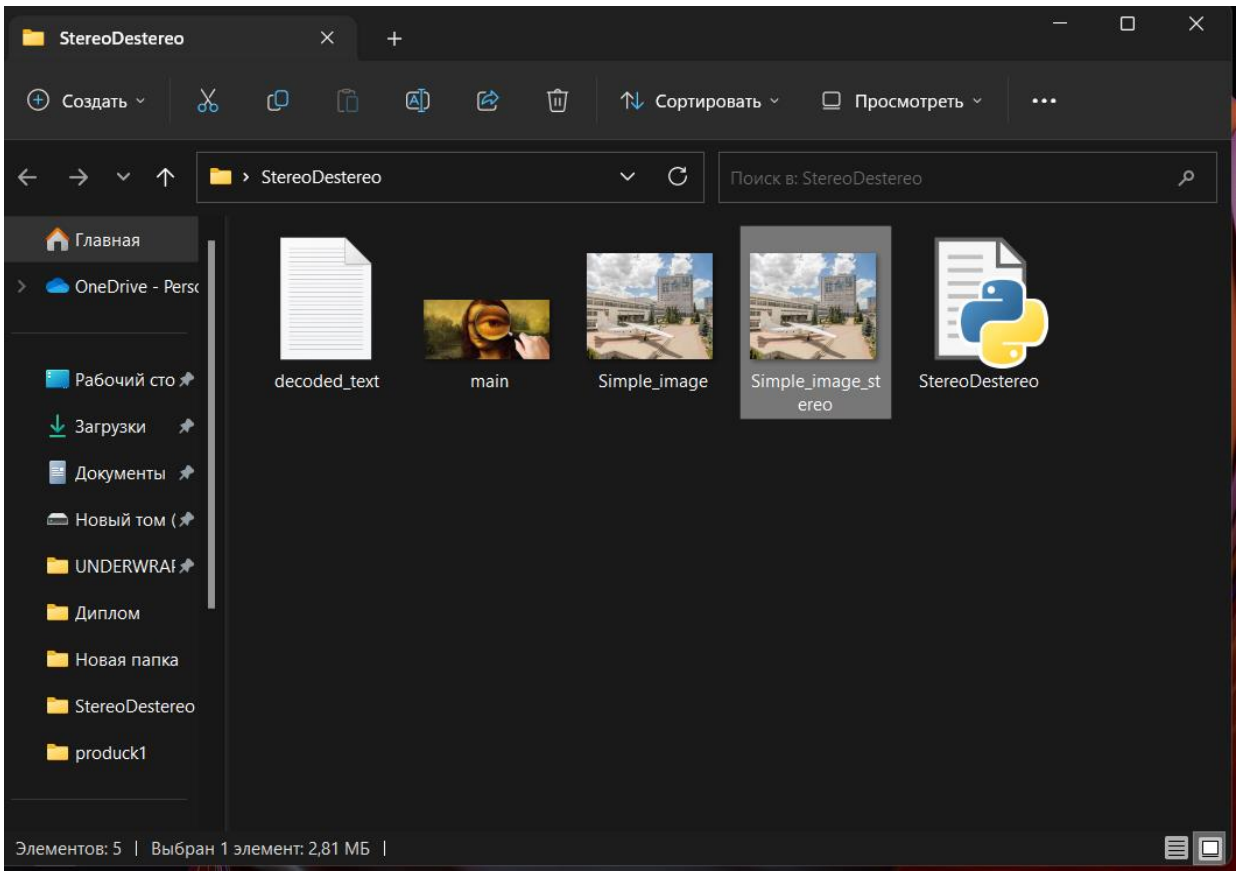


Рис.3.19. Створене нове зображення з зашифрованою інформацією

Декодування повідомлення: При натисканні кнопки "Декодувати", обране зображення піддається обробці. Застосовується метод LSB для вилучення прихованого тексту з зображення. Отриманий текст відображається у вікні разом з відповідним підписом. Результат розшифровки показав на рис.3.20.

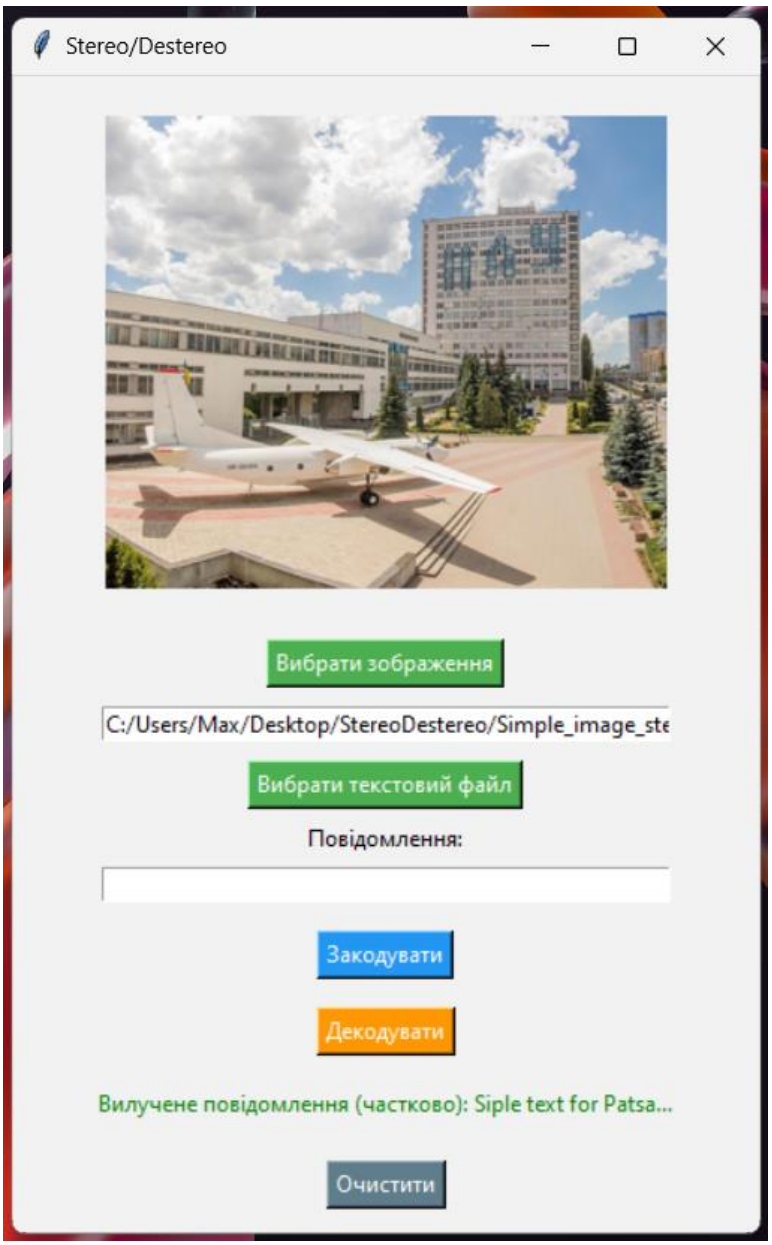


Рис.3.20. Повідомлення про успішне виконання

Збереження декодованого повідомлення: Після успішного декодування повідомлення, користувач має можливість зберегти його у текстовому файлі. Перші 20 символів декодованого повідомлення відображаються на екрані, а повне повідомлення зберігається у текстовому файлі з тією ж назвою, що й закодоване зображення, але з розширенням ".txt" рис.3.21.

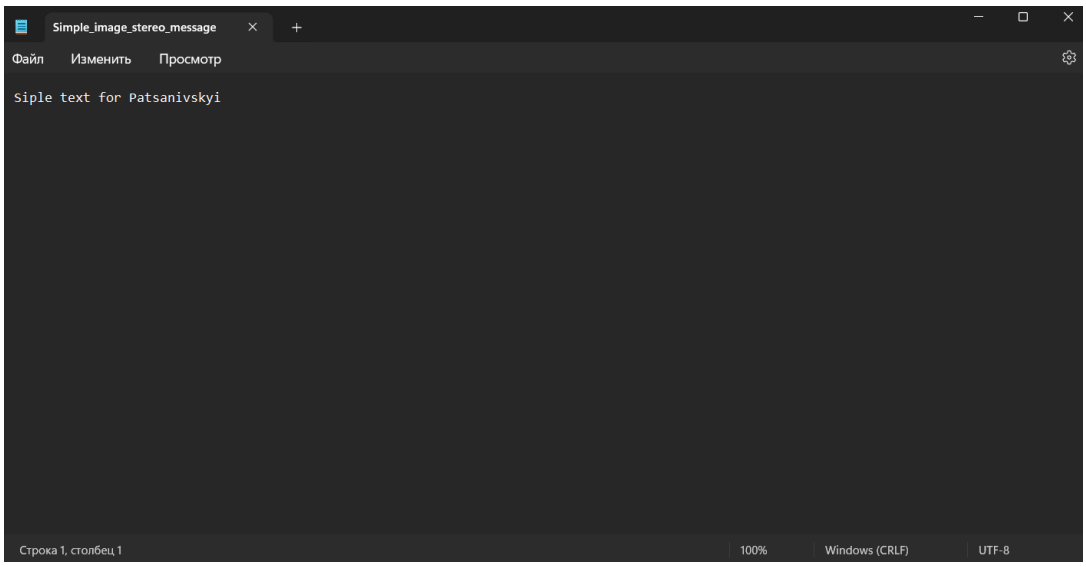


Рис.3.21. Збережене повідомлення у текстовому файлі

Очищення: Кнопка "Очистити" видаляє всі дані та скидає застосунок до початкового стану, готового для нових операцій. Звіт кнопки "Очистити" на рис.3.22.

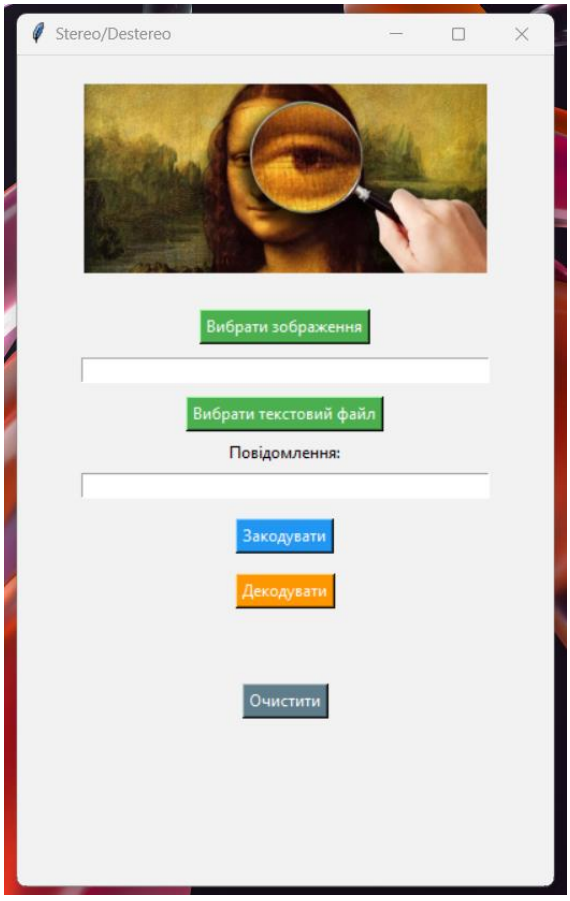


Рис.3.22. Звіт кнопки "Очистити"

Автоматичний режим: Якщо користувач не вибрав зображення або текстовий файл, але натиснув кнопку "Закодувати", застосунок автоматично запропонує вибрати відповідні файли, щоб уникнути зайвих дій. Можемо переглянути на рис.3.23. про наявність такого повідомлення, яке виділено червоним текстом.

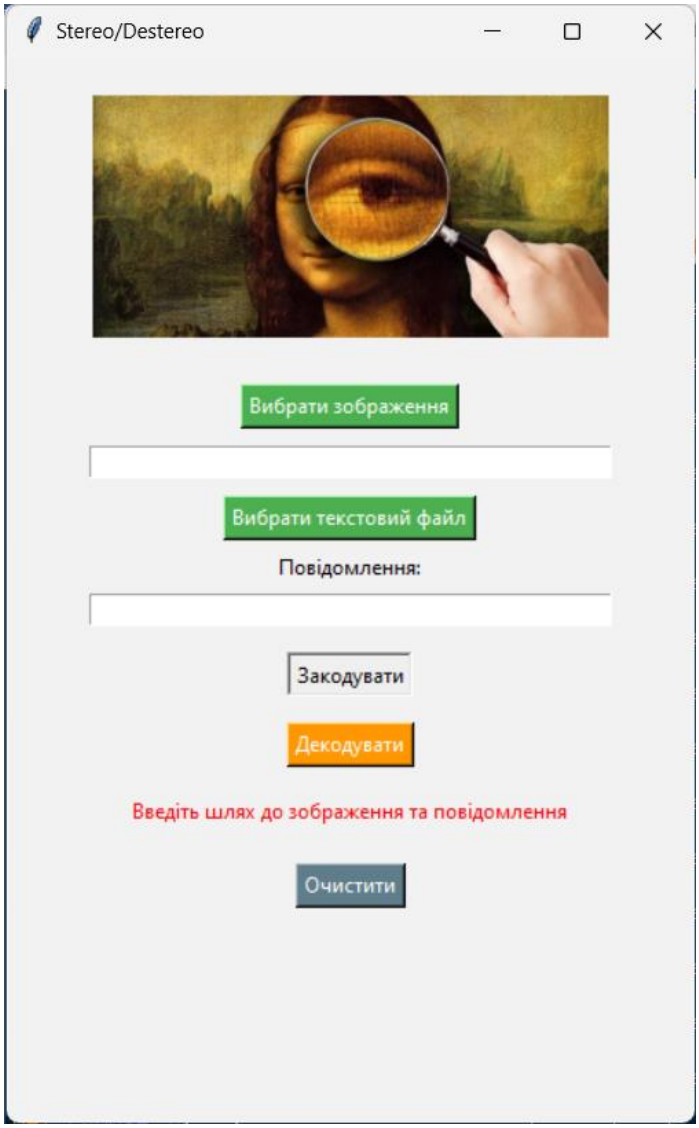


Рис.3.23. Помикла про невідповідність шляху чи повідомлення

Цей функціонал дозволяє користувачам ефективно використовувати цифрову стеганографію для захисту та обміну конфіденційною інформацією. Застосунок забезпечує зручний і простий у використанні інтерфейс, а також надійний алгоритм кодування і декодування для забезпечення безпеки і точності.

Реалізація програми на основі цифрової стеганографії з використанням мови програмування Python:

```
from tkinter import *
from tkinter import filedialog
from PIL import Image
from stegano import lsb
from PIL import ImageTk
import os

def hide_message(image_path, message):
    try:
        img = Image.open(image_path)
        encoded_img = lsb.hide(img, message)
        encoded_image_path = image_path.replace(".bmp", "_stereo.bmp")
        encoded_img.save(encoded_image_path)
        result_label.config(text="Повідомлення успішно вбудоване в зображення", fg="green")
    except Exception as e:
        result_label.config(text="Помилка: " + str(e), fg="red")

def extract_message(image_path):
    try:
        img = Image.open(image_path)
        hidden_message = lsb.reveal(img)
        result_label.config(text=f"Вилучене повідомлення: {hidden_message[:]}", fg="green")
                # Збереження розкодованого повідомлення у текстовому файлі
        output_file_path = image_path.replace(".bmp", "_message.txt")
        with open(output_file_path, "w", encoding="utf-8") as file:
            file.write(hidden_message)

        result_label.config(text=f"Вилучене повідомлення (частково): {hidden_message[:]}", fg="green")
    except Exception as e:
        result_label.config(text="Помилка: " + str(e), fg="red")

def encode_button_clicked():
    image_path = image_entry.get()
    message = message_entry.get()

    if image_path == "" or message == "":
        result_label.config(text="Введіть шлях до зображення та повідомлення", fg="red")
        return
```

```
    hide_message(image_path, message[:] + "")
    result_label.config(text=f"Закодоване повідомлення:
{message[:]}...", fg="green")

def decode_button_clicked():
    image_path = image_entry.get()

    if image_path == "":
        result_label.config(text="Введіть шлях до зображення",
fg="red")
        return

    extract_message(image_path)

def choose_image():
    image_path = filedialog.askopenfilename(filetypes=[("BMP Images", "*.bmp")])
    if image_path:
        image_entry.delete(0, END)
        image_entry.insert(0, image_path)
        display_image(image_path)

def display_image(image_path):
    image = Image.open(image_path)
    image.thumbnail((300, 300))
    photo = ImageTk.PhotoImage(image)
    image_label.config(image=photo)
    image_label.image = photo

def clear_button_clicked():
    clear_fields()
    display_image("main.jpg")

def clear_fields():
    image_entry.delete(0, END)
    message_entry.delete(0, END)
    result_label.config(text="", fg="black")
    image_label.config(image=None)

def choose_text_file():
    text_file_path = filedialog.askopenfilename(filetypes=[("Text Files", "*.txt")])
    if text_file_path:
        with open(text_file_path, 'r', encoding='utf-8') as file:
            text = file.read()
            message_entry.delete(0, END)
            message_entry.insert(0, text)
```

```
encoded_image_path = ""

# Створення графічного інтерфейсу
root = Tk()
root.title("Stereo/Destereo")
root.geometry("400x620")

# Фоновий колір
root.configure(bg="#f2f2f2")

# Зображення
image_label = Label(root, bg="#f2f2f2")
image_label.pack(pady=20)

choose_image_button = Button(root, text="Вибрати зображення",
command=choose_image, bg="#4CAF50", fg="white")
choose_image_button.pack(pady=5)

# Шлях до зображення
image_entry = Entry(root, width=50)
image_entry.pack(pady=5)

# Повідомлення
choose_text_button = Button(root, text="Вибрати текстовий файл",
command=choose_text_file, bg="#4CAF50", fg="white")
choose_text_button.pack(pady=5)
message_label = Label(root, text="Повідомлення:", bg="#f2f2f2")
message_label.pack()

message_entry = Entry(root, width=50)
message_entry.pack(pady=5)

# Кнопки
encode_button = Button(root, text="Закодувати",
command=encode_button_clicked, bg="#2196F3", fg="white")
encode_button.pack(pady=10)

decode_button = Button(root, text="Декодувати",
command=decode_button_clicked, bg="#FF9800", fg="white")
decode_button.pack(pady=5)

# Результат
result_label = Label(root, text="", bg="#f2f2f2")
result_label.pack(pady=10)
```

```
clear_button = Button(root, text="Очистити", command=clear_button_clicked,
bg="#607D8B", fg="white")
clear_button.pack(pady=10)

# Відображення початкового зображення "main.jpg" при запуску
display_image("main.jpg")

root.mainloop()
```

Крім того, код додатку може бути подальшим розширеним та вдосконаленим. Наприклад, можна додати можливість вибору інших форматів зображень, підтримку шифрування даних для більшої безпеки або покращення графічного інтерфейсу для поліпшення користувальницького досвіду.

Застосування цифрової стеганографії може мати широкий спектр застосувань, від захисту конфіденційної інформації у різних сферах, таких як військова та фінансова, до художнього використання для створення унікальних та цікавих ефектів зображень. Цей додаток є лише одним з численних прикладів застосування стеганографії та демонструє, наскільки сильним та потужним інструментом вона може бути.

У цілому, розробка та демонстрація додатку Stereo/Destereo на основі цифрової стеганографії показала, як стеганографія може бути використана для приховування та вилучення інформації зображень. Цей проект підкреслює важливість захисту конфіденційної інформації та демонструє, наскільки сильним та ефективним інструментом може бути цифрова стеганографія.

3.7. Висновок до розділу 3

У даному розділі було розглянуто застосування цифрової стеганографії з використанням мови програмування Python. Проведено огляд мови Python та розглянуто її можливості, які роблять її ідеальним інструментом для реалізації алгоритмів стеганографії.

Також була розглянута побудова стегоконтейнерів за допомогою зображень у форматі BMP. Зображення у цьому форматі використовуються як носії для приховання повідомлень. Розглянуті алгоритми приховування файлів методом "заміни найменших значущих бітів" та їх відтворення, які були реалізовані у мові програмування Python. Ці алгоритми дозволяють ефективно вбудовувати інформацію у зображення, забезпечуючи високий рівень захисту і надійності інформації.

Далі були представлені алгоритми видобування файлів із стегоконтейнерів у мові Python. Ці алгоритми дозволяють витягти приховану інформацію зі зображень, що була вбудована за допомогою методу "заміни найменших значущих бітів". Розглянуто використання методу Least Significant Bit (LSB) у мові Python, який є одним із найпоширеніших методів для приховування інформації.

Надалі, була реалізована та продемонстрована програмна програма на основі цифрової стеганографії з використанням мови програмування Python. Цей застосунок демонструє можливості стеганографії для приховування повідомлень у зображеннях, а також видобування цих повідомлень зі стегоконтейнерів.

Застосування цифрової стеганографії з використанням мови програмування Python відкриває широкі перспективи для безпечного обміну конфіденційною інформацією та забезпечує високий рівень захисту даних. Мова програмування Python є потужним інструментом для реалізації алгоритмів стеганографії, оскільки вона поєднує в собі зручний синтаксис, багатий набір бібліотек та велику спільноту розробників.

Отже, застосування цифрової стеганографії з використанням мови програмування Python є актуальним та перспективним напрямом досліджень. Реалізація алгоритмів стеганографії у мові Python дозволяє забезпечити високу ефективність та надійність процесу приховування та видобування інформації. Подальші дослідження у цій галузі можуть сприяти розвитку інформаційної безпеки та забезпеченню захисту конфіденційної інформації.

ВИСНОВКИ

У дипломному проекті була розроблено застосунок Stereo/Destereo, який використовує передові технології цифрової стеганографії для надійного та безпечного приховування та вилучення інформації зображень. Застосунок має інтуїтивний та зручний інтерфейс, що дозволяє користувачам легко вибирати зображення та текстові файли, а також здійснювати операції кодування та декодування з натисканням лише кількох кнопок.

Основні функції додатку включають можливість вибору зображення та текстового файлу для кодування, процес закодування текстових даних у вибране зображення за допомогою методу LSB, відображення закодованого зображення разом з підписом, що позначає закодований текст, можливість збереження закодованого зображення для подальшого використання або поділу з іншими, а також процес декодування прихованої інформації з обраного зображення.

Розроблений додаток Stereo/Destereo виявився ефективним і зручним у використанні, надаючи користувачам зручні та безпечні засоби для приховування та вилучення інформації зображень. Цей додаток може знайти своє застосування в різних сферах, де збереження конфіденційності та захист даних є важливими аспектами.

Застосунок був успішно протестований та готовий до загального використання користувачами.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Neil F. Johnson. Information Hiding: Steganography and Watermarking - Attacks and Countermeasures / Zoran Duric, Sushil Jajodia – Springer, Kluwer Academic Publishers, 2000. – 220 с.
2. Jessica Fridrich. Steganography in Digital Media: Principles, Algorithms, and Applications / Cambridge University Press, 2009. – 500 с.
3. Luca Lista. Statistical Methods for Data Analysis in Particle Physics / Springer, 2015. – 257 с.
4. Ніл Ф. Джонсон. Стеганографія: відкриваючи невидиме. / Сушіл Джаджодія – Київський університет, 2004. – 192 с.
5. Джессіка Фрідріх. Стеганографія в цифрових медіа: принципи, алгоритми та застосування. / Вид-во Наш Формат, 2012. – 376 с.
6. Інгемар Дж. Кокс. Цифрове водяне позначення і стеганографія, 2-е видання / Меттью Л. Міллер, Джеффрі А. Блум, Джессіка Фрідріх, Тон Калкер – Вид-во БХВ-Петербург, 2009. – 752 с.
7. Jessica Fridrich. Digital Watermarking and Steganography: Fundamentals and Techniques / Miroslav Goljan, Dorin Hogea. – Вид-во CRC Press, 2017. – 420 с.
8. Raval. Deep Learning for Computer Vision / Amit. – Вид-во Packt Publishing, 2017. – 300 с.
9. Хорошко В.О. Основи комп'ютерної стеганографії: Навч. посіб. для студентів і аспірантів. / Азаров О.Д., Шелест М.Є., Яремчук Ю.Є. – Вінниця: ВДТУ, 2003 – 155 с.
10. Cryptography and Steganography [Електронний ресурс] – Режим доступу: https://www.cs.sjsu.edu/~stamp/steganography. (Дата звернення 23.05.2023) – Назва з екрану.
11. The International Association for Cryptologic Research [Електронний ресурс] – Режим доступ: https://iacr.org. (Дата звернення 23.05.2023) – Назва з екрану.

12. Steganography in Python [Електронний ресурс] – Режим доступу: https://github.com/ragibson/Steganography. (Дата звернення 28.05.2023) – Назва з екрану.

13. Steganography in Images using Python [Електронний ресурс] – Режим доступу: https://www.geeksforgeeks.org/steganography-in-images-using-python. (Дата звернення 29.05.2023) – Назва з екрану.

14. Steganography in Python [Електронний ресурс] – Режим доступу: https://medium.com/swlh/steganography-in-python-4b62b56fc329. (Дата звернення 29.05.2023) – Назва з екрану.

15. Катсоссідіс Й. "Стеганографія і стеганоаналіз". / Київ: Національний технічний університет України "КПІ", 2010. – 292 с.

16. Жерар Е. "Стеганографія: Технології та застосування". / Париж: HERMES, 2007. – 272 с.

17. Хусен Х. А. "Стеганографія: методи та застосування". / Харків: Видавничий дім "ІНЖЕК", 2012. – 200 с.

18. Холі М. "Стеганографія та цифрова водяна печатка". / Київ: Національний університет "Києво-Могилянська академія", 2009. – 148 с.