**Міністерство освіти і науки України**
**Національний технічний університет**
**«Дніпровська політехніка»**

**Інститут електроенергетики**
**Факультет інформаційних технологій**
**Кафедра безпеки інформації та телекомунікацій**

# ПОЯСНЮВАЛЬНА ЗАПИСКА
## кваліфікаційної роботи ступеня магістра

**студента** *Дзеркаля Романа Артурович*

**академічної групи** *125-22-2*

**спеціальності** *125 Кібербезпека*

**спеціалізації**[1]

**за освітньо-професійною програмою** *Кібербезпека*

**на тему** *Дослідження особливостей Python бібліотек в процесі використання методів цифрової стеганографії*

| **Керівники** | **Прізвище, ініціали** | **Оцінка за шкалою** | | **Підпис** |
|---|---|---|---|---|
| | | **рейтинговою** | **інституційною** | |
| кваліфікаційної роботи | д.т.н., проф. Корнієнко В. І. | 80 | добре | |
| розділів: | | | | |
| спеціальний | ст. викл. Тимофєєв Д.С. | 80 | добре | |
| економічний | к.е.н., доц. Пілова Д. П. | 95 | відмінно | |

| | | | | |
|---|---|---|---|---|
| **Рецензент** | к. т. н., доц. Шидловський І. А. | 80 | добре | |

| | | | | |
|---|---|---|---|---|
| **Нормоконтролер** | ст. викл. Мєшков В. І. | 85 | добре | |

**Дніпро**
**2023**

**ЗАТВЕРДЖЕНО:**
завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.

«_____»_____2023 року

# ЗАВДАННЯ
## на кваліфікаційну роботу ступеня магістра

**студенту** *Дзеркалю Р. А.* **академічної групи** *125-22м-2*
(прізвище та ініціали) (шифр)

**спеціальності** *125 Кібербезпека*

**спеціалізації**

**за освітньо-професійною програмою** *Кібербезпека*

**на тему** *Дослідження особливостей Python бібліотек в процесі використання методів цифрової стеганографії*

Затверджену наказом ректора НТУ «Дніпровська політехніка» від 09.10.2023 № 1227-с

| **Розділ** | **Зміст** | **Термін виконання** |
|---|---|---|
| Розділ 1 | Аналіз використання цифрової стеганографіїї. Аналіз існуючих методів цифрової стеганографії та методів їх стегоаналізу. | 02.11.2023 |
| Розділ 2 | Дослідження Python бібліотек та існуючих ріщень; їх сумісність. Аналіз бібліотек, що реалізують методи стегоаналізу | 17.11.2023 |
| Розділ 3 | Визначення економічної доцільності та ефекту від дослідження Python бібліотек | 29.11.2023 |

Завдання видано _____ Тимофєєв Д.С.
(підпис керівника) (прізвище, ініціали)
Дата видачі завдання: 01.09.2023

Дата подання до екзаменаційної комісії: 06.12.2023

Прийнято до виконання _____ Дзеркаль Р. А.
(підпис студента) (прізвище, ініціали)

# РЕФЕРАТ

Пояснювальна записка: 57 с., 26 рис., 6 додатків, 13 джерел.

Об’єктом дослідження виступають алгоритми цифрової стеганографії різних типів контейнерів.

Предметом дослідження є бібліотеки Python для стеганографіїї та стегоаналізу.

Мета роботи: дослідження Python бібліотек у сфері цифрової стеганографії, порівняння та аналіз ефективності їх використання при розробці програмного забезпечення.

Методи дослідження: спостереження, порівняння, аналіз, опис.

У першому розділі було проаналізовано існуючі методи стеганографічного захисту інформації для рійних типів контейнерів, проаналізовано методи їх стеганографічного аналізу; сформульована та поставлена задача дослідження.

У спеціальній частині були розглянуті також існуючі програмні засоби, що реалізують алгоритми стеганографічного захисту інформації; було проведено дослідження Python бібліотек та проаналізовано доцільність їх сумісного використання.

У економічному розділі розраховано витрати на проведення дослідження та визначено економічний ефект.

PYTHON БІБЛІОТЕКИ, МЕТОДИ СТЕГАНОГРАФІЧНОГО ЗАХИСТУ, МЕТОДИ СТЕГОАНАЛІЗУ, LSB, DWT, DST

ABSTRACT

Explanatory note: 57 pages., 26 figures, 1 table, 6 appendices, 13 sources.

The object of research is digital steganography algorithms of various types of containers.

The subject of research is Python libraries for steganography and steganalysis.

The purpose of the work: research of Python libraries in the field of digital steganography, comparison and analysis of the effectiveness of their use in software development.

Research methods: observation, comparison, analysis, description.

In the first section, the existing methods of steganographic protection of information for various types of containers were analyzed, the methods of their steganographic analysis were analyzed; research task formulated and set.

In a special part, existing software tools that implement algorithms for steganographic protection of information were also considered; a study of Python libraries was conducted and the expediency of their combined use was analyzed.

In the economic section, the costs of conducting the research are calculated and the economic effect is determined.

PYTHON LIBRARIES, METHODS OF STEGANOGRAPHIC PROTECTION, METHODS OF STEGOANALYSIS, LSB, DWT, DST

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

**LSB** – List Significant Bit

**HAS** – Human Auditory System

**ПЗ** – програмне забезпечення

**ІС** – інформаційні системи

**ІБ** – інформаційна безпека

**TCO** – Total Cost Ownership

**ROI** - Return of investments

**ROSI** – Return of security investments

**PVD** - Pixel Value Differencing

**HVS** – Human Vision System

**SS** – Spread Spectrum

**HEVC** - High Efficiency Video Coding

**CTU** – Coding Tree Unit

**DCT** – Discrete cosine transform

**DST** - Discrete sinus transform

**GOP** – Group of Picture

**DWT** - Discrete Wavelet Transform

**HTML** – Hyper Text Markup Language

**XML** - Extensible Markup Language

**JPEG** - Joint Photographic Experts Group

**RGB** - red, green, blue

**BMP** - bitmap

ЗМІСТ

# ВСТУП

Об’єктом дослідження виступають алгоритми цифрової стеганографії різних типів контейнерів.

Предметом дослідження є бібліотеки Python для стеганографіїї та стегоаналізу.

Мета роботи: дослідження Python бібліотек у сфері цифрової стеганографії, порівняння та аналіз ефективності їх використання при розробці програмного забезпечення.

Розвиток цифрових технологій та засобів телекомунікацій призвів до того, що інформація стала все більш доступною та уразливою для несанкціонованого доступу. Це підвищує ризик витоку конфіденційної інформації. Наразі для захисту інформації від несанкціонованого доступу існує два основних підходи: шифрування та цифрова стеганографія.

Перший підхід полягає у блокуванні від несанкціонованого доступу до інформації методом перетворення (або шифрування) повідомлення. Для цього використовуються різні криптографічні методи, які у свою чергу поділяються на симетричні та асиметричні.

Другий підхід полягає у приховані самого факту існування цінної інформації. Для цього використовуються стеганографічні методи захисту, які значно знижують ймовірність її виявлення. Щоб не можна було навіть запідозрити існування переданої інформації. У такому випадку віднайти цільову інформацію стає важким завданням. Але попри це стеганографія часто використовується у поєднанні з методами криптографічного шифрування, щоб збільшити надійність передачі інформації.

У цифровій стеганографії існує таке поняття як «контейнер» - цифровий об'єкт, у якому вбудовується секретне повідомлення. Контейнер може бути будь-яким файлом: зображенням, відео, аудіо, текстовим документом, скомпресованим файлом тощо. Вибір контейнера є важливим фактором у проектуванні стеганографічної системи.

# РОЗДІЛ 1. СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ

## 1.1 Стан питання

Стеганографія – це мистецтво прихованого письма, метою якої є прихована комунікація. Це відрізняється від криптографії, мистецтва секретного письма, яке має на меті зробити повідомлення недоступним для прочитання третій стороні, але не обов'язково приховує саме існування таємного спілкування.

Стеганографія використовується для приховання цільового повідомлення . Попри це, стеганографія також може використовуватися незаконними методами як-от: приховування записів про незаконну діяльність, фінансове шахрайство, промислове шпигунство та спілкування між членами злочинних або терористичних організацій. Таким чином, стеганографія представляє інтерес для дипломатичних, військових, розвідувальних, кримінальних, терористичних і правоохоронних органів – для передачі таємних повідомлень або для їх виявлення.

Перші згадки терміну «стеганографія» сягають кінця XV століття, але практичне використання стеганографії налічує кілька тисяч років. Після винайдення письма (приблизно 2000 р. до н. е.), людство навчилося писати секретними кодами, а невдовзі – й таємно спілкуватися. Так, наприклад, у стародавні часи, послання ховали на звороті воскових таблиць; писали на животі кроликів або татуювали шкіру голови рабів, використовуючи невидимі чорнила.

Стеганографія приховує приховане повідомлення, але не обов'язково щоб дві сторони спілкуються між собою. В епоху цифровізації, приховувати повідомлення можна у будь-якому бінарному вигляді: електронній пошті, фотографіях, відео або музиці тощо – у так званих «контейнерах». Найрозповсюдженіший контейнер на сьогодні є саме графічні зображення.

### 1.1.2 Стеганографія у відеоіграх

Розробники ігор також використовують стеганографію для ідентифікації автора скріншотів або ігрових відео, особливо коли вони містять шахрайство, зловживання або несанкціоноване використання приватних серверів.

У 2000-х роках компанія Blizzard впровадила дуже слабкі водяні знаки на скріншотах World of Warcraft, які містили повторювані візерунки з точок по всьому екрану. Ці візерунки, розроблені Digimarc, кодували різні деталі облікового запису користувача та сервера, на якому він був зареєстрований. Як і інші наведені вище приклади, це тегування скріншотів залишалося цілком таємним протягом перших кількох років свого існування (рис. 1.1.1.1).

Рис. 1.1 Два приклади водяних знаків, що використовуються у World of Warcraft, підданих значній постобробці, щоб виявити прихований шаблон

Подібним чином Microsoft зашифрувала інформацію про апаратне забезпечення в інтерфейсі користувача ранніх версій Xbox 360. Анімація кожної консолі була унікальною, що дозволяло компанії боротися з потенційними витоками. У той час співробітники компанії були підпорядковані договорам про нерозголошення, і за розголошення непублічної інформації про розробку консолі їм загрожували цивільно-правові штрафи.

Ігрові приставки дозволяють бандам і терористичним організаціям спілкуватися на міжнародному рівні, уникаючи при цьому викриття. Обвинуваченим, засудженим до домашнього арешту часто забороняють користуватися комп'ютером для доступу до інтернету. Однак, якщо ігрові приставки не заборонені, правопорушник все одно має можливість виходити в інтернет [6].

1.1.3 Поняття водяного знаку

Стеганографія має таке поняття як «цифровий водяний знак». Це дозволяє автор вставити приховане повідомлення у файл, щоб пізніше підтвердити своє право на інтелектуальну власність та забезпечити цілісність вмісту. Художник,

наприклад, може розмістити на веб-сайті якийсь оригінальний твір мистецтва. Якщо хтось інший «вкраде» файл і заявить, що твір є його/її власністю, художник може пізніше підтвердити право власності, тому що тільки він/вона може відновити водяний знак. Хоча концептуально подібний до стего, цифровий водяний знак зазвичай має інші технічні цілі, а саме:

- Як правило, водяний знак – це невелика кількість повторюваної інформації, яка вставляється на носій
- Не завжди необхідно приховувати інформацію водяних знаків від глядача
- Це корисно, якщо водяний знак можна видалити, зберігаючи цілісність файлу-носія.

1.2 Методи стеганографічного захисту інформації

У цьому розділі розглядаються стеганографічні методи маскування даних для різних типів контейнерів. При цьому значна увага приділяється проблемі практичної реалізації розглянутих методів з використанням сучасних засобів обчислювальної техніки та програмного забезпечення. Також будуть розглянуті методи стеганографічного аналізу для обраних контейнерів.

Під час розгляду методів будемо позначати буквою С контейнер, що являє собою послідовність елементів із довжиною Ic. У випадку використання в якості контейнера файлу цифрового звуку, це буде кількість відліків за одиницю часу, для файлу цифрового зображення - послідовність, отримана шляхом векторизації зображення (тобто, шляхом розгортання масиву всіх пікселів зображення у вектор).

Для двійкових масивів контейнерів значення c1 можуть приймати значення "0" або "1"; для квантованого зображення або звуку (якщо кількість біт, яким кодується один відлік, дорівнює 8) - змінюватися в діапазоні від 0 до 255 (2^8 = 256 градацій); для звуку, відліки якого кодуються 16-ма бітами, в діапазоні від -32768 до 32767 (2^16 = 65536 градацій); для текстів c1 - це символ кодової таблиці, код якого може приймати значення від 0 до 255.

Аналогічно, будемо позначати буквою S заповнений контейнер (стеганограму) послідовність елементів S1 довжиною Ls, а буквою M -

повідомлення довжиною Lm, яке підлягає маскуванню. Якщо не буде особливо обумовлено інше, будемо вважати, що M = {0; 1}.

Класифікація методів приховування даних

Переважна більшість методів комп'ютерної стеганографії (КС) базується на двох ключових принципах:

1. файли, які не вимагають абсолютної точності (наприклад, файли з зображенням, звуковою інформацією і т.д.), можуть бути видозмінені (звичайно, до певної міри) без втрати своєї функціональності;
2. органи чуття людини не здатні надійно розрізняти незначні зміни в модифікованих таким чином файлах і/або відсутній спеціалізований інструментарій, який був би здатний виконувати цю задачу.

В основі базових підходів до реалізації методів КС в рамках тієї чи іншої інформаційної середовища лежить виділення малозначущих фрагментів цієї середовища і заміна існуючої в них інформації інформацією, яку необхідно приховати. Оскільки в КС розглядаються середовища, підтримувані засобами обчислювальної техніки і комп'ютерних мереж, то вся інформаційна середовище в результаті може бути представлена в цифровому вигляді.

Таким чином, незначні для кадра інформаційної середовища фрагменти відносно того чи іншого алгоритму або методики замінюються (заміщуються) фрагментами прихованої інформації. Під кадром інформаційної середовища в даному випадку мається на увазі певна його частина, виділена по характерним ознакам. Такими ознаками часто є семантичні характеристики виділеної частини інформаційної середовища. Наприклад, кадром може бути обрано якесь окреме зображення, звуковий файл, веб-сторінка і т.д.

Для існуючих методів комп'ютерної стеганографії можливі наступні класифікації.

За способом вибору інформації:

- Суррогатні методи — інформація замінюється на малозначущі елементи контейнера.

- Селекційні методи — інформація замінюється на елементи контейнера, які не впливають на його сприйняття.
- Конструкторські методи — інформація вбудовується в контейнер таким чином, що її неможливо виявити.

За способом доступу до інформації:

- З методами з оригіналом — інформація може бути витягнута без зміни контейнера.
- З методами без оригіналу — інформація може бути витягнута тільки після зміни контейнера.

За способом організації контейнера:

- Потокові методи — інформація вбудовується в контейнер послідовно.
- Систематичні методи — інформація вбудовується в контейнер за певним алгоритмом.

За форматом контейнера:

- Текстові методи — інформація вбудовується в текстовий контейнер.
- Графічні методи — інформація вбудовується в графічний контейнер.
- Аудіо методи — інформація вбудовується в аудіо контейнер.
- Відео методи — інформація вбудовується в відео контейнер.

В суррогатних (безальтернативних) методах стеганографії повністю відсутня можливість вибору контейнера, і для скриття повідомлення вибирається перший попавшийся контейнер, эрзац-контейнер, який в більшості випадків не оптимальний для скриття повідомлення заданого формату.

В селективних методах КС передбачається, що приховане повідомлення має відтворювати спеціальні статистичні характеристики шуму контейнера. Для цього генерують велику кількість альтернативних контейнерів з подальшим вибором (путем відбраковки) найбільш оптимального з них для конкретного повідомлення. Особливим випадком такого підходу є обчислення деякої хеш-функції для кожного контейнера. При цьому для скриття повідомлення вибирається той контейнер, хеш-функція якого збігається зі значенням хеш-функції повідомлення (тобто стеганограмою є обраний контейнер).

В конструюючих методах стеганографії контейнер генерується самою стеганосистемою. При цьому існує кілька варіантів реалізації. Так, наприклад, шум контейнера може імітуватися прихованим повідомленням. Це реалізується за допомогою процедур, які не тільки кодують приховане повідомлення під шум, але і зберігають модель изначального шуму. В предельному випадку по моделі шуму може будуватися ціле повідомлення. Прикладом може служити метод, реалізований в програмі MandelSteg, яка в якості контейнера генерує Фрактал Мандельброта (Mandelbrot fractal), або ж апарат функції імітації

По способу доступу до прихованої інформації розрізняють методи для поточних (безперервних) контейнерів і методи для фіксованих (обмеженої довжини) контейнерів (більш докладно див. розділ 2.3).

По способу організації контейнери, подібно завадостійким кодам, можуть бути систематичними і несистематичними. В перших можна вказати конкретні місця стеганограми, де знаходяться інформаційні біти власне контейнера, а де - шумові біти, призначені для скриття інформації (як, наприклад, в широко поширеному методі найменшого значущого біта).

В випадку несистематичної організації контейнера таке розділення неможливо. В цьому випадку для виділення прихованої інформації необхідно обробляти зміст всієї стеганограми.

За принципом приховування методи комп'ютерної стеганографії поділяються на два основних класи: методи безпосередньої заміни та спектральні методи.

Якщо перші, використовуючи надлишок інформаційного середовища в просторовій (для зображення) або часовій (для звуку) області, полягають у заміні малозначущої частини контейнера бітами секретного повідомлення, то інші для приховування даних використовують спектральні представлення елементів середовища, в яку впроваджуються приховувані дані (наприклад, у різні коефіцієнти масивів дискретно-косинусних перетворень, перетворень Фурье, Карунена-Лоева, Адамара, Хаара тощо) [5].

Основним напрямком комп'ютерної стеганографії є використання властивостей саме надлишку контейнера-оригіналу, але при цьому слід

враховувати те, що в результаті приховування інформації відбувається спотворення деяких статистичних властивостей контейнера або ж

В окрему групу можна також виділити методи, які використовують спеціальні властивості форматів подання файлів:

- зарезервовані для розширення поля файлів, які часто заповнюються нулями і не враховуються програмою;
- спеціальне форматування даних (зсув слів, речень, абзаців або вибір певних позицій символів);
- використання незадіяних ділянок на магнітних і оптичних носіях: видалення файлових заголовків-ідентифікаторів тощо.

Тож, для таких методів характерні низька ступінь прихованості, низька пропускна здатність і слабка продуктивність.

За призначенням розрізняють стеганометоди власне для прихованої передачі (або прихованого зберігання) даних і методи для приховування даних у цифрових об'єктах з метою захисту авторських прав на них.

Розглянемо докладніше стеганографічні методи приховування даних для кожного із зазначених типів контейнеру.

Для початку розглянемо методи приховування інформації у зображеннях

Зображення є найпопулярнішим контейнером для стеганографії. У сфері цифрових зображень існує багато різних форматів файлів зображень і для них різних форматів файлів зображень існують різні стеганографічні алгоритми. [8]

Методи стеганографії зображень можна розділити на дві групи: ті, що належать до домену зображення, і ті, що належать до домену трансформації. Методи просторової області зображення вбудовують повідомлення безпосередньо в інтенсивність пікселів, тоді як для трансформації, відомої також як частотна область, зображення спочатку перетворюються, а потім повідомлення вбудовується в зображення.

Методи домену зображення охоплюють побітові методи, які застосовують бітову вставку та маніпулювання шумом і іноді характеризуються як «прості

системи». Формати зображень, які найбільше підходять для стеганографії домену зображення, є без втрат, а методи зазвичай залежать від формату зображення.

Стеганографія в області перетворення передбачає маніпулювання алгоритмами та перетвореннями зображень. Ці методи приховують повідомлення в більш значущих областях зображення обкладинки, роблячи його більш надійним. Багато методів домену трансформації не залежать від формату зображення, і вбудоване повідомлення може пережити перетворення між стисненням із втратами та без втрат. У наступних розділах стеганографічні алгоритми будуть пояснені в категоріях відповідно до форматів файлів зображень і домену, в якому вони виконуються

Розглянемо домен зображення

### 1.2.1 LSB

Вставка молодшого значущого біта (LSB) є поширеним простим підходом до вбудовування інформації в зображення обкладинки [14]. Молодший біт (іншими словами, 8-й біт) деяких або всіх байтів усередині зображення змінюється на біт секретного повідомлення. При використанні 24-розрядного зображення, трохи кожного з червоного, зеленого та синього кольорів можна використовувати компоненти, оскільки кожен з них представлений байтом. Іншими словами, в кожному пікселі можна зберігати 3 біти. Таким чином, зображення розміром 800 x 600 пікселів може зберігати загальну кількість 1 440 000 біт або 180 000 байт вбудованих даних [19]. Наприклад, сітка для 3 пікселів 24-бітного зображення може бути такою:

(00101101 00011100 11011100)
(10100110 11000100 00001100)
(11010010 10101101 01100011)

Коли число 200, двійкове представлення якого дорівнює 11001000, вставляється в молодші біти цієї частини зображення, результуюча сітка має такий вигляд:

(00101101 00011101 11011100)

(10100110 11000101 00001100)

(11010010 10101100 01100011)

Хоча число було вбудовано в перші 8 байтів сітки, лише 3 підкреслені біти потрібно було змінити відповідно до вбудованого повідомлення. У середньому потрібно буде змінити лише половину бітів зображення, щоб приховати секретне повідомлення, використовуючи максимальний розмір обкладинки. Оскільки існує 256 можливих інтенсивностей кожного основного кольору, зміна LSB пікселя призводить до невеликих змін інтенсивності кольорів. Ці зміни не можуть бути сприйняті людським оком - таким чином повідомлення успішно приховано. З добре підібраним зображенням можна навіть приховати повідомлення в найменших, а також від другого до молодшого значущих бітів і все одно не побачити різниці.

У наведеному вище прикладі для вбудовування інформації використовуються послідовні байти даних зображення - від першого байта до кінця повідомлення. Такий підхід дуже легко виявити. Трохи безпечніша система полягає в тому, що відправник і одержувач мають спільний секретний ключ, який визначає лише певні пікселі, які потрібно змінити. Якщо зловмисник підозрює, що була використана стеганографія LSB, він не зможе дізнатися, які пікселі націлити, без секретного ключа.

У своїй найпростішій формі LSB використовує зображення BMP, оскільки вони використовують стиснення без втрат. На жаль, щоб мати можливість приховати секретне повідомлення у файлі BMP, знадобиться дуже велике зображення обкладинки. В даний час BMP-зображення розміром 800 x 600 пікселів не часто використовуються в Інтернеті і можуть викликати підозру. З цієї причини стеганографія LSB також була розроблена для використання з іншими форматами файлів зображень.

Наступним доменом буде домен трансформації

Щоб зрозуміти алгоритми стеганографії, які можна використовувати під час вбудовування даних у домен трансформації, потрібно спочатку пояснити тип формату файлу, пов’язаного з цим доменом. Формат файлу JPEG є

найпопулярнішим форматом файлу зображень в Інтернеті через малий розмір зображень.

### 1.2.2 JPEG стискання

Щоб стиснути зображення у формат JPEG, подання кольору RGB спочатку перетворюється на подання YUV. У цьому поданні компонент Y відповідає яскравості (або яскравості), а компоненти U і V означають кольоровість (або колір). Згідно з дослідженнями, людське око більш чутливе до змін яскравості (яскравості) пікселя, ніж до змін його кольору. Цей факт використовується під час стиснення JPEG шляхом зменшення роздільної здатності даних кольорів для зменшення розміру файлу. Колірні компоненти (U і V) зменшуються вдвічі в горизонтальному і вертикальному напрямках, таким чином зменшуючи розмір файлу в 2 рази.

Наступний крок – власне трансформація образу. Для JPEG використовується дискретне косинусне перетворення (DCT), але подібними перетвореннями є, наприклад, дискретне перетворення Фур'є (DFT). Ці математичні перетворення перетворюють пікселі таким чином, щоб отримати ефект «розповсюдження» розташування піксельних значень на частині зображення. DCT перетворює сигнал із представлення зображення в представлення частоти шляхом групування пікселів у піксельні блоки 8x8 і перетворення піксельних блоків у 64 коефіцієнти DCT кожен Модифікація одного коефіцієнта DCT вплине на всі 64 пікселя зображення в цьому блоці.

Наступним кроком є фаза квантування стиснення. Тут використовується інша біологічна властивість людського ока: людське око досить добре помічає невеликі відмінності в яскравості на відносно великій площі, але не настільки добре, щоб розрізняти різну силу високочастотної яскравості. Це означає, що інтенсивність вищих частот може бути зменшена без зміни зовнішнього вигляду зображення. JPEG робить це шляхом ділення всіх значень у блоці на коефіцієнт квантування. Результати округлюються до цілих значень, а коефіцієнти кодуються за допомогою кодування Хаффмана для подальшого зменшення розміру. Розглянемо детальніше JPEG стеганографія.

Спочатку вважалося, що стеганографію неможливо використовувати із зображеннями JPEG, оскільки вони використовують стиснення з втратами, що призводить до зміни частини даних зображення. Однією з основних характеристик стеганографії є той факт, що інформація прихована в надлишкових бітах об'єкта, а оскільки зайві біти пропускаються під час використання JPEG, існує побоювання, що приховане повідомлення буде знищено. Навіть якби можна було якимось чином зберегти повідомлення недоторканим, було б важко вставити повідомлення без помітних змін через застосоване жорстке стиснення. Проте властивості алгоритму стиснення були використані для розробки стеганографічного алгоритму для JPEG.

Одна з цих властивостей JPEG використовується для того, щоб зробити зміни в зображенні невидимими для людського ока. Під час фази перетворення DCT алгоритму стиснення в даних коефіцієнта виникають помилки округлення, які не помітні. Хоча ця властивість є тим, що класифікує алгоритм як втрачений, цю властивість також можна використовувати для приховування повідомлень.

Вбудовувати інформацію в зображення, яке використовує стиснення з втратами, неможливо, оскільки стиснення знищить всю інформацію в процесі. Таким чином, важливо визнати, що алгоритм стиснення JPEG насправді поділяється на етапи з втратами та без втрат. DCT і фаза квантування є частиною етапу з втратами, тоді як кодування Хаффмана, яке використовується для подальшого стиснення даних, є без втрат. Стеганографія може відбуватися між цими двома етапами. Використовуючи ті самі принципи вставки LSB, повідомлення можна вставити в молодші біти коефіцієнтів перед застосуванням кодування Хаффмана. Вбудовуючи інформацію на цьому етапі в область трансформації, її надзвичайно важко виявити, оскільки вона не знаходиться у візуальній області.

Розглянемо наступні домени: домен трансформації і домен зображення.

### 1.2.3 Patchwork

Patchwork — це статистичний метод, який використовує надлишкове кодування шаблонів для вбудовування повідомлення в зображення. Алгоритм

додає надмірність до прихованої інформації, а потім розсіює її по всьому зображенню. Псевдовипадковий генератор використовується для вибору двох областей зображення (або патчів), патча A та патча B. Усі пікселі в ділянці A освітлені, тоді як пікселі в ділянці B затемнені. Іншими словами, інтенсивність пікселів в одному патчі збільшується на постійне значення, тоді як пікселі іншого патча зменшується на таке ж постійне значення. Зміни контрасту в цій підмножині патчів кодують один біт, і зазвичай зміни невеликі та непомітні, але не змінюють середню яскравість.

Недоліком підходу печворк є те, що вбудовано лише один біт. Можна вбудувати більше бітів, спочатку розділивши зображення на підзображення та застосувавши вбудовування до кожного з них. Перевага використання цієї техніки полягає в тому, що секретне повідомлення розподіляється по всьому зображенню, тому якщо один патч буде знищено, інші можуть ще вижити. Однак це залежить від розміру повідомлення, оскільки повідомлення може повторюватися на всьому зображенні, лише якщо воно досить маленьке. Якщо повідомлення занадто велике, його можна вставити лише один раз.

Цей підхід використовується незалежно від зображення хоста та виявляється досить надійним, оскільки приховане повідомлення може витримати перетворення між стисненням із втратами та без втрат.

### 1.2.4 Розширення спектру

У методах розширеного спектру приховані дані розподілені по всьому зображенню, що ускладнює виявлення.

Зв'язок із розширеним спектром можна визначити як процес розширення смуги пропускання вузькосмугового сигналу в широкому діапазоні частот. Це можна досягти, регулюючи вузькосмугову форму хвилі за допомогою широкосмугової форми, наприклад, білого шуму. Після розширення енергія вузькосмугового сигналу в будь-якій смузі частот є низькою, і тому її важко виявити. У стеганографії зображення з розширеним спектром повідомлення вбудовується в шум, а потім поєднується із зображенням обкладинки для

отримання стегозображення. Оскільки потужність вбудованого сигналу набагато нижча за потужність зображення обкладинки, вбудоване зображення не сприймається людським оком або комп'ютерним аналізом без доступу до оригінального зображення.

Розглянемо наступні методи для текстового типу контейнеру.

Текстова стеганографія [3] може включати в себе все, що завгодно, від зміни форматування існуючого тексту, зміни слів у тексті, генерації випадкових послідовностей символів або використання контекстних граматик випадкових послідовностей символів або використання контекстно-вільних граматик для створення читабельних текстів. Текстова стеганографія вважається найскладнішою через відсутність надлишкової інформації, яка присутня у зображенні, аудіо або відеофайлі. Структура текстових документів ідентична тому, що ми спостерігаємо. Непомітні зміни можуть бути внесені до зображення або аудіофайлу, але в текстових файлах навіть додаткова буква або знак пунктуації може бути помічена випадковим читачем.

При вкладенні даних у текстовий файл основною проблемою є його структура, яка не повинна змінюватися. Якщо структура змінюється, вся суть текстового файлу змінюється, тоді як в інших цифрових носіях зміни можна легко зробити без внесення помітних змін до відповідного виводу.

Зберігання текстового файлу вимагає менше пам'яті, а його більш швидка і легка передача робить його кращим над іншими типами стеганографічних методів. Текстову стеганографію можна умовно розділити на три типи: форматна, випадкова та статистична генерація, лінгвістичні методи.

Форматні методи засновані на модифікації форматування тексту, наприклад, на додаванні додаткових пробілів або символів. Такі методи прості у реалізації, але мають обмежену пропускну здатність. В оригінальному повідомленні не змінюються слова і речення; модифікації будуть внесені лише в проміжки між словами, рядками та/або абзацами за допомогою спеціальних символів, тобто стеганографії білих проміжків.

Випадкові та статистичні методи генерують випадковий текст, у який потім вбудовують секретне повідомлення. Такі методи більш надійні, ніж форматні, але вимагають більше обчислювальних ресурсів.

Лінгвістичні методи використовують особливості мови для приховування секретного повідомлення. Такі методи можуть бути дуже ефективними, але вимагають глибокого знання мови. Використовується для приховування повідомлення в іншому повідомленні залежно від мовної структури прихованого повідомлення (пунктуації) або семантики слів як місця для приховування повідомлення. Методи текстової стеганографії також ще можна класифікувати на два типи:

1) Зміна формату тексту: у цьому методі формат текстового файлу змінюється таким чином, щоб приховати секретне повідомлення. Це можна зробити шляхом зміни розширення файлу, вставлення додаткових пропусків між словами або символів, яких не видно під час друку.
2) Зміна значення тексту: цей метод фокусується на зміні самого тексту для приховування секретного повідомлення. Це можна зробити шляхом заміни слів синонімами, переставляння слів або зміни граматики.

Розглянемо деякі з популярних підходів до стеганографії тексту. [4]

### 1.2.4 Зсув рядка

У цьому методі прихований повідомлення ховається шляхом вертикального зсуву рядків тексту на деякий ступінь. Щоб приховати біт 0, рядок зсувається вгору, а щоб приховати біт 1, рядок зсувається вниз. Визначення того, чи був рядок зсунутий вгору чи вниз, здійснюється шляхом вимірювання відстані між центром позначеної лінії та її контрольними лініями. Якщо використовується програма розпізнавання символів (OCR), прихована інформація буде знищена. Крім того, відстані можна спостерігати за допомогою спеціальних приладів для оцінки відстані.

### 1.2.5 Зсув слова

У цьому методі приховане повідомлення ховається шляхом зсуву слів горизонтально, тобто вліво або вправо, щоб представляти біт 0 або 1 відповідно. Зсув слів детектується за допомогою методу кореляції, який розглядає профіль як хвилеформу та вирішує, чи вона походить із хвилеформи, середина блоку якої була зміщена вліво або вправо. Цей метод може бути менш ідентифікованим, оскільки зміна відстані між словами для заповнення рядка досить поширена [10, 11]. Однак, якщо хтось знає алгоритм відстаней, він може порівняти стегонований текст з алгоритмом і отримати прихований вміст за допомогою різниці.

### 1.2.6 Синтаксичний метод

Ця техніка використовує розділові знаки, такі як крапка (.), кома (,) тощо, для приховування бітів 0 і 1. Але проблема з цим методом полягає в тому, що він вимагає ідентифікації правильних місць для вставки розділових знаків. Тому при використанні цього методу слід бути обережними, оскільки читачі можуть помітити неправильне використання розділових знаків.

### 1.2.7 Білий стеганограф

Ця техніка використовує прогалини для приховування секретного повідомлення. Існує три методи приховування даних за допомогою прогалин. У методі інтервалів між реченнями ми розміщуємо одну прогалину, щоб приховати біт 0 і дві прогалини, щоб приховати біт 1 в кінці кожного кінцевого символу. У методах кінців рядків фіксована кількість прогалин вставляється в кінці кожного рядка. Наприклад, дві прогалини для кодування одного біту на рядок, чотири прогалини для кодування двох бітів і так далі. У методиці інтервалів між словами одна прогалина після слова представляє біт 0, а дві прогалини після слова представляють біт 1.

### 1.2.8 Спам-текст

HTML- і XML-файли також можна використовувати для приховування бітів. Якщо є різні початкові та кінцеві теги, то біт 0 інтерпретується, а якщо для початку та закриття використовується один тег, то інтерпретується біт 1. У іншій техніці біт

0 представляється відсутністю проміжку в тезі, а біт 1 представляється розміщенням проміжку в тезі.

1.2.9 Стеганографія в XML-документах

Виконання стеганографії на XML-документах є ефективним, оскільки він широко використовується для обміну даними, а також вважається мовою цифрового вмісту та веб-сторінок. Техніка приховує дані шляхом вставки випадкових символів між тегами XML та їх значеннями. Ця техніка відома як техніка випадкових символів. Вставка випадкових символів збільшується від 1 до n після кожного слова тега. Цей процес повторюється доки не зустрінеться крапка (.). Потім цей процес застосовується рекурсивно до всіх тегів у документі XML. Друга техніка виконує перемішування тегів, що відбувається в заздалегідь визначеній послідовності. При цьому 1-й тег міняється місцями з останнім тегом, 2-й тег з передостаннім тегом. Та сама процедура повторюється, доки не будуть переставлені всі теги. Міняються місцями як позиція, так і значення тега. Третя процедура відома як перемішування тегів, визначене атрибутом, яка зберігає порядок тегів в атрибутах перед перемішуванням. Остання техніка відома як зворотна символічна техніка, в якій послідовність символів у тезі обертається, наприклад, якщо тег "width", то він буде обернений як "htdiw". Після обернення тега його значення також обертається. Процедура повторюється доки не зустрінеться крапка (.)

Далі розглянемо методи стеганографії для відео контейнера.

Відеостеганографія [5] — це процес приховування секретної інформації всередині відео. Секретною інформацією може бути будь-який медіа-файл, як-от текст, аудіо, зображення, відео та двійковий файл, а відео-носій може бути у будь-якому форматі, сирому чи стиснутому. Детальна класифікація методів відеостеганографії за різними критеріями наведена на рис. 1.2.3.1. Перший рівень класифікації базується на форматі відео-носія. Відео-носії розглядаються або в raw області, або в стиснутій області. Відео raw домену додатково класифікується на просторову область і область перетворення. Заміна найменш значущих бітів (LSB)

та інші важливі методи включені до просторової області. Дискретне вейвлетне перетворення (DWT) і дискретне косинусне перетворення (DCT) є широко використовуваними методами перетворення для перетворення відео-носіїв у область трансформації. Після перетворення відео в область трансформації здійснюється вбудовування секретної інформації. У стиснутій області відеостеганографія використовує стиснуті відео-носії. Методи відеостеганографії, що широко використовуються в стиснутій області, включають вектори руху, режими внутрішньокадрового прогнозування, модулі кодування ентропії та техніки DCT/DST.

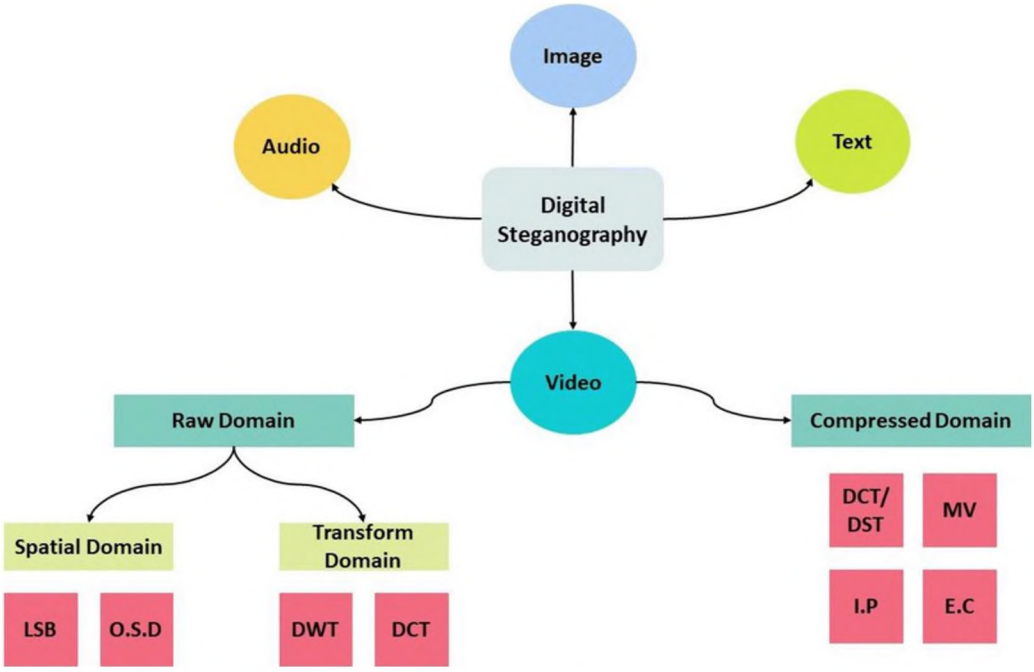


Рис. 1.2 - Класифікація методів стеганографії відеоконтейнеру

Для початку розглянемо методи приховування даних у області трансформації

На відміну від методу на основі просторової області, метод на основі домену перетворення перетворює блоки кадрів. Після цього секретні дані вбудовуються в молодші біти коефіцієнтів перетворення. Загальний робочий процес приховування даних в області перетворення показаний на рис. 1.2.3.1.1. Дискретне вейвлет-перетворення (DWT) і дискретне косинусне перетворення (DCT) є двома функціями перетворення, які переважно використовуються у відеостеганографії.

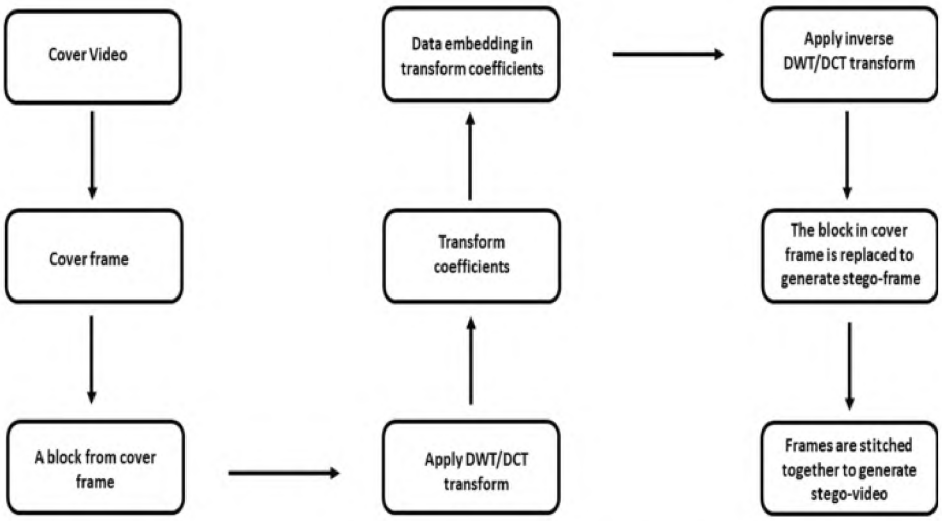


Рис. 1.3 - Процес приховування даних в області перетворення

1.2.10 Дискретне вейвлет-перетворення

Дискретне вейвлетне перетворення (DWT) розкладає сигнал на набори з суттєвою та незначущою інформацією. Важлива інформація пов'язана із загальним виглядом і називається низькочастотними коефіцієнтами DWT. Подібним чином, незначна інформація представляє поведінку сигналів і називається високочастотними коефіцієнтами. Один сигнал проходить через набір фільтрів і розкладається на дві частини - наближення і деталі. Рядки та стовпці зображення r × c передаються та обробляються незалежно. Формула, яка використовується для розкладання рядків і стовпців, наведена в (1) і (2) відповідно.

$$i(x,y) = \begin{cases} \sum_0^{n-1} I(r,m).h_L(m-r), r \equiv 0(mod2) \\ \sum_0^{n-1} I(r,m).h_H(m-r), r \equiv 1(mod2) \end{cases} \quad (1)$$

$$i'(x,y) = \begin{cases} \sum_0^{n-1} i(m,c).h_L(m-c), c \equiv 0(mod2) \\ \sum_0^{n-1} i(m,c).h_H(m-c), c \equiv 1(mod2) \end{cases} \quad (2)$$

де I(r,c) – зображення, hL – фільтр низьких частот, а hH – фільтр високих частот.

Нарешті, низькочастотні компоненти розташовані у верхній половині, тоді як високочастотні компоненти розташовані в нижній половині. Ті самі кроки повторюються протягом кількох ітерацій.

1.2.11 Дискретне косинусне перетворення

DCT також є функцією перетворення, як DWT, яка розділяє зображення на спектральні піддіапазони. Основна відмінність полягає в тому, що попередній генерує більше частотних діапазонів і забезпечує вищу роздільну здатність за частотою. Тим не менш, DWT генерує кілька діапазонів частот і забезпечує високу просторову роздільну здатність. Значна кількість робіт у літературі використовувала домен DWT для вбудовування секретних даних у необроблені відео. На відміну від DWT, домен DCT не часто використовується в літературі для приховування секретних даних у необроблених відео. З іншого боку, методи стеганографії відео, запропоновані в стисненому домені, широко використовують домен DCT для приховування секретних даних. У цьому розділі розглядається метод на основі DCT лише у домені необробленого відео. DCT застосовується до кожного кадру відео окремо та перетворює кадр на низькі, середні та високочастотні смуги. Секретні дані приховані в коефіцієнтах перетворення одного або кількох діапазонів.

Розглянемо довільний фрейм $I$ із роздільною здатністю $J \times K$. А $T$ — це перетворений фрейм, згенерований застосуванням DCT до $I$. Коефіцієнти DCT для $T$ обчислюються за допомогою рівняння:

$$T_{xy} = \alpha_x \alpha_y \sum_{j=0}^{J-1} \sum_{k=0}^{K-1} I_{jk} \cos\frac{\pi(2j+1)x}{2J} \cos\frac{\pi(2k+1)y}{2k} \quad (3)$$

Після вбудовування бітів секретних даних у коефіцієнти DCT до кадру T застосовується інверсний двовимірний DCT, щоб створити кадр I за допомогою рівняння

$$I_{jk} = \sum_{x=0}^{J-1} \sum_{y=0}^{K-1} \alpha_x \alpha_y T_{xy} \cos\frac{\pi(2j+1)x}{2J} \cos\frac{\pi(2k+1)y}{2k} \quad (4)$$

Де

$$\alpha_x = \begin{cases} \frac{1}{\sqrt{\frac{2}{j}}}, & x = 0 \\ \sqrt{\frac{2}{j}}, & 1 \le x \le J-1 \end{cases} \quad (5)$$

$$\alpha_y = \begin{cases} \frac{1}{\sqrt{K}}, & y = 0 \\ \sqrt{\frac{2}{K}}, 1 \le x \le JK - 1 \end{cases} \quad (6)$$

Тут $I_{jk}$ представляє значення пікселів у клітинці (стовпець $j$ рядок $K$) кадру $I$.

Далі розглянемо приховування даних у області стискання

Більшість запропонованих у літературі методів відеостеганографії для приховування даних у необробленому домені прості та легкі у реалізації. Але, вони більш схильні до різних атак, особливо атак стиснення. Крім того, наразі відео в стиснутому вигляді є кращим для зберігання та передачі через Інтернет. Стиснене відео потребує менше місця для зберігання, ніж нестисне відео. А передавання відео в стиснутому вигляді через Інтернет відбувається швидше та потребує меншої пропускної здатності. У цьому контексті методи приховування даних у домені стисненого відео набули популярності за останні два десятиліття. З іншого боку, стиснення призводить до видалення зайвих відеоданих і зменшує простір для приховування додаткових даних.

Серед різноманітних доступних стандартів кодування стиснення відео MPEG-X і H.26X є широко використовуваними методами в останні роки. Зокрема, стандарт відеокодування H.264/AVC, також відомий як MPEG-4 Part-10. Відеокодек H.264 має кілька нових функцій порівняно зі своїми попередниками, а деякі з нових функцій – це «можливість посилань на кілька кадрів», «гнучке впорядкування макроблоків», внутрішньокадрове передбачення тощо. Загалом відеокодек H.264 складається з кілька груп картинок (GOP). Кожна GOP містить кадри з внутрішнім кодуванням (I-кадр), прогнозовані кадри (P-кадр) і двонаправлені прогнозовані кадри (B-кадр). I-кадр, він же ключовий кадр, є незалежно закодованим і першим кадром кожної GOP. P-кадр містить лише різницю між поточним кадром і попереднім кадром. B-кадр зберігає лише зміни в поточному кадрі з попередніх і наступних кадрів.

Процедура кодування відео в кодеку H.264 показана на рис. 1.4. Під час кодування початковий кадр (який містить усі важливі дані та вважається I-кадром) ділиться на макроблоки, де кожен макроблок складається з 16 × 16 пікселів. .

Процес стиснення даних включає різні етапи, такі як прогнозування, перетворення домену та кодування. Прогноз використовує часову та просторову надмірність у відеоданих. Прогнозування дозволяє кодувати різницю між попередньо закодованими та прогнозованими даними. Існує два типи передбачення: внутрішнє передбачення та міжпередбачення. Внутрішнє передбачення генерує передбачення макроблоків на основі попередньо закодованих даних у поточному кадрі, тоді як інтерпрогнозування генерує передбачення на основі даних у попередньо закодованих кадрах. Для прогнозування кадру використовуються методи оцінки руху та компенсації руху. Різниця між прогнозом і поточним макроблоком називається залишком. Блок залишків піддається доменному перетворенню за допомогою цілочисельного перетворення. DCT є найбільш часто використовуваним цілим перетворенням. Блок перетворених коефіцієнтів квантується, щоб мінімізувати точність коефіцієнтів.

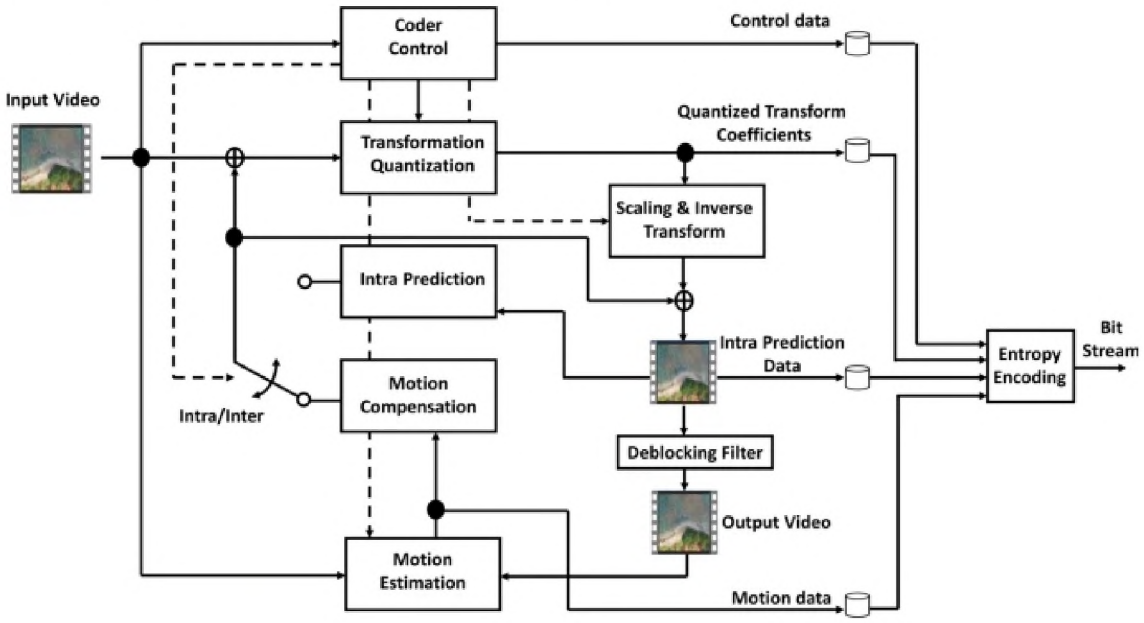


Рис. 1.4 – Блок-схема функціонування кодеку H.264

Останній крок кодування перетворює різні значення (квантовані коефіцієнти DCT, дані, необхідні декодеру для реконструкції передбачення, інші дані про відеопослідовність тощо), отримані на попередніх етапах, і елементи синтаксису в двійкові коди.

У стислому домені приховування даних реалізовано двома способами; приховування даних разом із процедурою кодування відео та приховування даних

у кодованому бітовому потоці. Методи приховування даних разом із процедурою кодування відео використовують різні синтаксичні елементи, пов'язані із завданням кодування відео для вбудовування секретних даних. Загальний огляд приховування даних в елементах синтаксису стиснутого домену представлено на рис. 1.5. Методи приховування даних у кодованому потокі бітів використовують модулі ентропійного кодування для перенесення секретних даних.

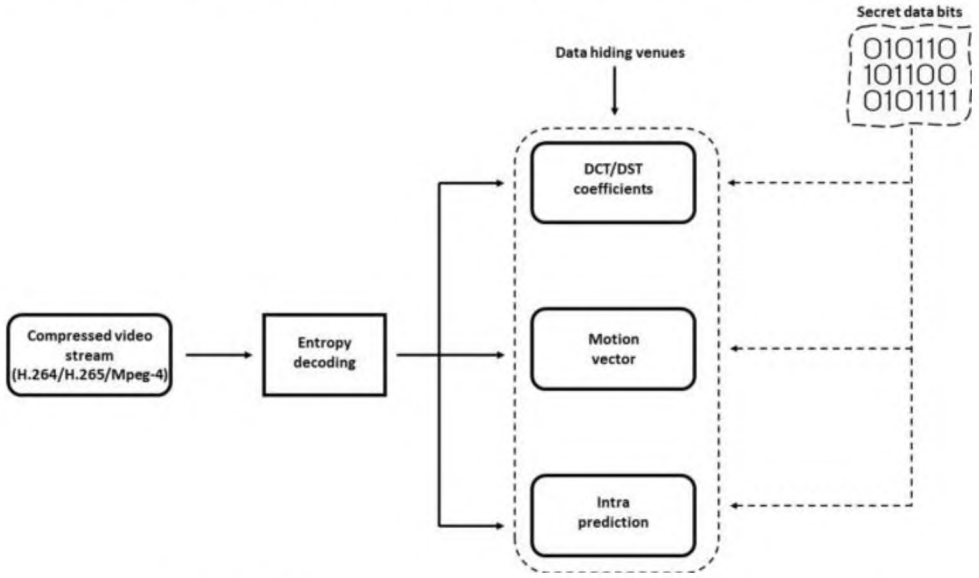


Рис. 1.5- Огляд приховування даних у домені стиснення

1.2.12 DCT/DST коефіцієнти

У літературі квантовані коефіцієнти DCT, отримані під час процедури кодування відео, використовувалися переважно для приховування секретних даних у відео H.264. Як правило, секретні дані вбудовані в коефіцієнти DCT блоку яскравості 4×4 кадру покриття (особливо I-кадру). Внутрішньокадрове спотворення є однією з головних проблем, з якими стикаються під час вбудовування секретних даних у стиснене відео. У внутрішньокадровому прогнозуванні поточний блок передбачення буде сумою залишкових значень і прогнозованих значень. Передбачуване значення обчислюється із вибірок блоку сусіднього закодованого блоку. Припустимо, що сусідній закодований блок є одним із місць для приховування секретних даних, має бути спотворення через вбудовування даних. Оскільки прогнозовані значення обчислюються з його сусідніх блоків, це спотворення, спричинене вбудовуванням у попередньому блоці,

поширюватиметься на прогнозований блок за допомогою внутрішньокадрового прогнозування.

Щоб впоратися зі спотворенням, спричиненим у відео через збільшення кількості секретних даних, прихованих у кожному блоці, вводиться схема збурення на основі DCT. У схемі збурення вибирається новий відфільтрований блок яскравості 4×4, щоб приховати секретні дані шляхом збурення відповідних квантованих коефіцієнтів DCT. Кількісна оцінка запропонованої схеми збурення на основі DCT показує, що здатність до вбудовування покращується без шкоди для візуальної якості. Щоб вставити секретні дані та запобігти дрейфу внутрішньокадрових спотворень, квантовані коефіцієнти DCT класифікуються у два різних кластери. Перший кластер зарезервовано для вбудовування секретних даних, тоді як другий кластер використовується для запобігання дрейфу спотворення всередині кадру. Представлено таблицю напрямків модифікації вбудовування для вбудовування секретних даних у коефіцієнти DCT з мінімальним викривленням. Спочатку створюється таблиця модифікації напрямку вбудовування для секретних бітів. Кожен елемент у таблиці модифікації напрямку вбудовування відповідає кожному біту секретного повідомлення. Потім значення напрямку модифікації вбудовування кластера, зарезервованого для вбудовування (вбудовуючий кластер), обчислюється за спеціальним рівнянням. Після цього обчислюється різниця між значенням модифікації напрямку вбудовування кластера вбудовування та десятковими значеннями n секретних бітів для регулювання вбудовування та спотворення.

H.265/HEVC (високоефективне кодування відео) — це найдосконаліший стандарт кодування відео наступного покоління, який може забезпечити більше стиснення, ніж H.264/AVC, не впливаючи на якість відео. Кодування H.265/HEVC включає дії, подібні до кодування H.264/AVC. Аналогічно макроблокам у H.264, одиниця дерева кодування (CTU) розміром до 64×64 пікселів є основною одиницею кодування в H.265/HEVC. CTU можна додатково розділити на кілька одиниць кодування розміром 32 × 32 пікселів, 16 × 16 пікселів, 8 × 8 пікселів, 4 × 4 пікселів, дотримуючись структури квадродерева. Крім того, HEVC використовує два різних

цілочисельних перетворення, які базуються на DCT і DST для кодування залишкових блоків.

1.2.13 Вектори руху

Вектор руху є важливим синтаксичним елементом у процесі кодування відео, і він використовується для оцінки руху, а також для компенсації руху, щоб зменшити часову надмірність. Техніка компенсації руху дозволяє передбачити поточний блок на основі попередніх або майбутніх блоків, враховуючи рух об'єктів у кадрі. Вектор руху широко використовується для вбудовування секретних даних у стислому відео. Більшість існуючих алгоритмів відеостеганографії на основі векторів руху можна класифікувати відповідно до підходу модифікації, який використовується у векторах руху для вбудовування даних. Вбудовування даних на основі модифікації величини вектора руху та модифікації фазового кута вектора руху є двома підходами. Підхід до вбудовування даних, заснований на модифікації величини, змінює значення величини відповідного вектора руху шляхом додавання або віднімання 1 на основі секретних бітів даних. Метод модифікації фазового кута перетворює декартову систему координат на уявну систему та розглядає кожну секцію уявних секцій як 0 або 1. На основі секретного біта, який потрібно приховати, вибрані вектори руху повертаються для вбудовування даних.

Розглянемо методи приховування інформації у аудіо контейнері

Особливої [6] важливості приховуванню даних в аудіофайлах надає їхня поширеність як носіїв інформації в нашому суспільстві. Обережна стеганографія передбачає, що контейнер, який використовується для приховування повідомлень, не повинен викликати підозр у конкурентів. Насправді, доступність та популярність аудіофайлів робить їх придатними для зберігання прихованої інформації. Крім того, більшість зусиль зі стеганалізу спрямовані на цифрові зображення, що робить стеганаліз аудіо відносно малодослідженою територією.

Приховування даних в аудіофайлах є особливо складним завданням через чутливість людського слуху. Проте, слух все ж таки терпить певні загальні зміни в

невеликих діапазонах. Наприклад, гучні звуки можуть заглушати тихі звуки. Крім того, існують деякі поширені навколишні спотворення, які в більшості випадків ігноруються слухачами. Ці властивості спонукали дослідників вивчати використання аудіосигналів як носіїв для приховування даних.

Розглянемо методи приховування в тимчасовому домені

1.2.14 Низькорозрядне кодування

Також відомий як LSB (молодший значущий біт), цей метод є одним із найперших методів, що використовувалися для приховування інформації. Традиційно він заснований на вбудовуванні кожного біта з повідомлення в найменш значущий біт обкладинки аудіо детермінованим способом. Таким чином, для аудіо з дискретизацією 16 кГц дані 16 Кбіт/с приховані. Метод LSB забезпечує високу ємність для вбудовування даних і його відносно легко реалізувати або поєднати з іншими методами приховування. Однак ця техніка характеризується низькою стійкістю до додавання шуму, що знижує ефективність її безпеки, оскільки вона стає вразливою навіть до простих атак. Фільтрація, підсилення, додавання шуму та стиснення стего-аудіо з втратами, швидше за все, знищать дані. Крім того, оскільки дані вбудовані дуже детермінованим способом, зловмисник може легко розкрити повідомлення, просто видаливши всю площину LSB. Хоч цей метод забезпечує непомітність при високій швидкості вбудовування, безпека та надійність прихованих даних легко можуть бути скомпрометовані.

1.2.15 Приховування відлуння

Метод приховування відлуння вбудовує дані в аудіосигнали шляхом введення короткого відлуння в сигнал. Природа луни – це резонанс, доданий до основного звуку. Таким чином, уникаються проблеми чутливості HAS до адитивного шуму. Після додавання луни стегосигнал зберігає ті самі статистичні та перцептивні характеристики. Дані приховуються шляхом маніпулювання трьома параметрами ехо-сигналу: початковою амплітудою, зсувом (затримкою) і швидкістю затухання, щоб відлуння не було чутно (рис. 1.6). Для затримки до 1 мс між вихідним сигналом і луною ефект нерозрізнити. На додаток до цього,

амплітуда та швидкість загасання можуть бути встановлені на значення нижче порогу чутності людського вуха. Таким чином, дані можуть бути приховані, не будучи помітними. Однак недоліком цього методу є обмеження розміру індукованого ехо-сигналу, що обмежує пов'язані з ним області застосування. Звідси обмежена кількість робіт, які досліджують застосування цього методу.

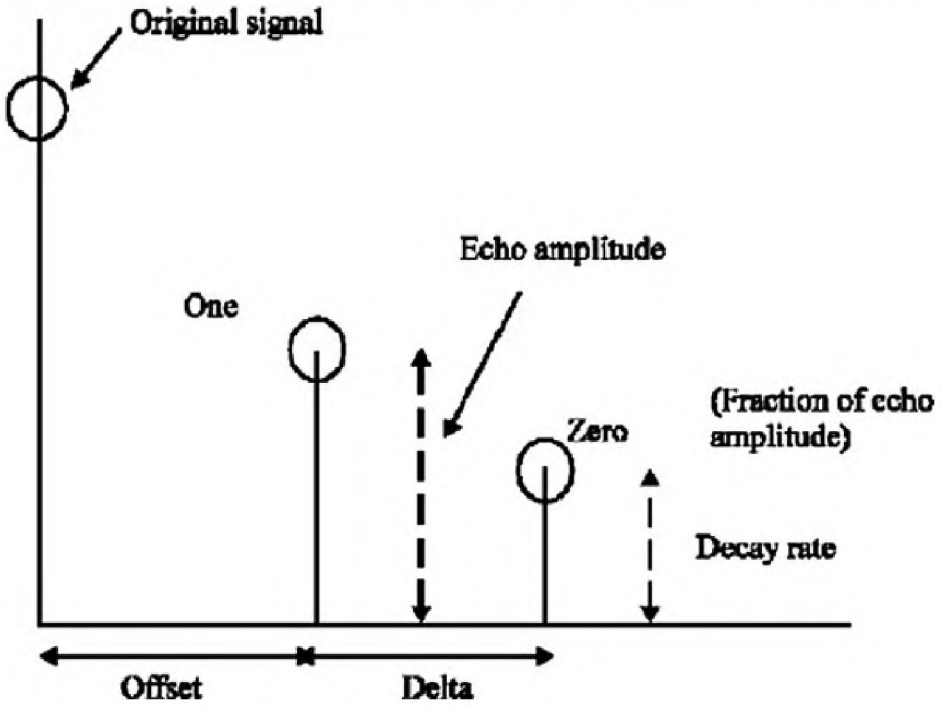


Рис. 1.6 – Приховування відлуння

1.2.16 Приховування у інтервалах тиші

Спочатку визначаються інтервали тиші та їх відповідна довжина (кількість зразків в інтервалі мовчання). Ці значення зменшуються на значення x, де x це кількість бітів, необхідних для представлення значення з повідомлення, яке потрібно приховати. Для процесу вилучення x оцінюється як mod(NewIntervalLength,$2^{nbits}$). Наприклад, якщо ми хочемо приховати значення 6 в інтервалі мовчання з довжиною 109, ми видаляємо 7 вибірок із цього інтервалу, що робить новий інтервал довжиною 102. Щоб отримати приховані дані з цього інтервалу тиші в стегосигналі, ми обчислюємо mod (102, 8) = 6. Невеликі інтервали тиші залишаємо незмінними, оскільки вони зазвичай трапляються в безперервних реченнях, і їх зміна може вплинути на якість мовлення. Цей метод має хорошу

прозорість сприйняття, але, очевидно, він чутливий до стиснення. Зміни в тривалості інтервалів мовчання призведуть до помилкового вилучення даних.

Розглянемо методи приховування в домені трансформації

1.2.17 Розширення спектру

Техніка розширення спектру поширює приховані дані через частотний спектр. Розширений спектр (SS) — це концепція, розроблена в сфері передачі даних, щоб забезпечити належне відновлення сигналу, надісланого по шумному каналу, шляхом створення надлишкових копій сигналу даних. По суті, дані помножуються на код М-послідовності, відомий як відправнику, так і одержувачу, а потім ховаються в обкладинці аудіо. Таким чином, якщо шум пошкоджує деякі значення, все одно залишаться копії кожного значення для відновлення прихованого повідомлення.

1.2.18 Вставка тону

Техніка вставки тону базується на нечутності тонів нижчої потужності за наявності значно вищих. Вбудовування даних шляхом вставки нечутних тонів у звукові сигнали покриття представлено в. Щоб вбудувати один біт у звуковий кадр, це дослідження пропонує пару тонів, які генеруються на двох вибраних частотах. Рівень потужності двох замаскованих частот встановлюється на відоме співвідношення загальної потужності кожного звукового кадру p i, де: i=1,…n i n — номер кадру, як показано на рисунку 1.7.

Вставляючи тональні сигнали на відомих частотах і на низькому рівні потужності, досягається приховане вбудовування та правильне вилучення даних. Для виявлення тонів і, отже, прихованої інформації з кадрів стего-аудіо, обчислюється потужність p i для кожного кадру, а також потужність для вибраних частот Якщо співвідношення, то прихований біт дорівнює «0», інакше — «1».

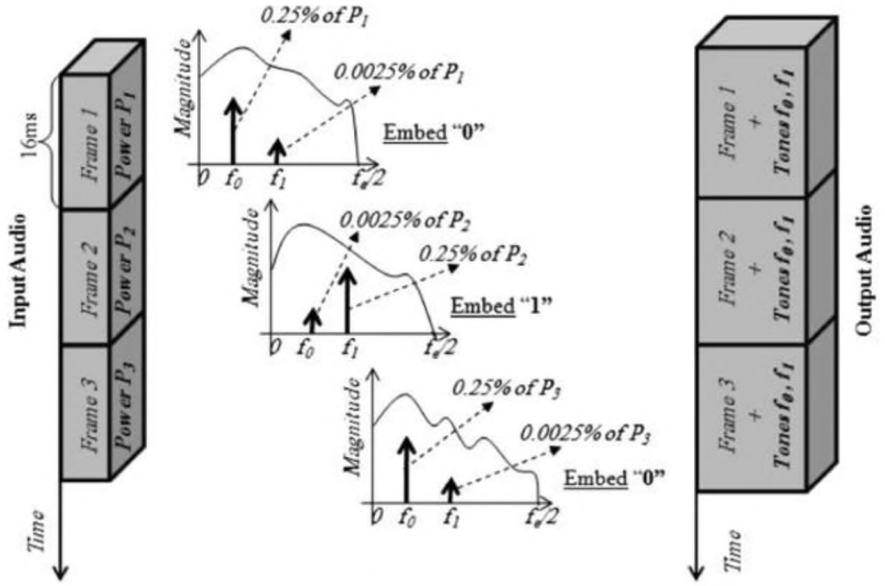


Рис. 1.7 - Вставка тону

## 1.3 Методи стеганографічного аналізу

Методи стеганалізу загалом поділяються на три типи: сигнатурний стеганаліз, статистичний стеганаліз і стеганаліз на основі ознак. Стеганаліз підпису, як випливає з назви, використовує підпис, залишений методом вбудовування, щоб виявити наявність секретної інформації. Статистичний стеганаліз використовує статистичні методи та математичні формули для виявлення та розкриття секретної інформації. Методи стеганалізу на основі функцій витягують ознаки з обкладинки відео та стего-відео для дослідження присутності та, таким чином, розкриття секретної інформації. Детальна діаграма розгалужень методів стеганалізу та їх підгалузей наведена на рис. 1.3.1.


Рис. 1.8 – Класифікация методів стегоаналізу

### 1.3.1 Сигнатурний стегоаналіз

Крім того, методи стеганалізу сигнатур поділяються на візуальні атаки та структурні атаки. Перша і головна риса, яка очікується від стеганографії — непомітність. Спотворення, викликані методом стеганографії, мають бути мінімальними, інакше сліди прихованого повідомлення можуть стати видимими для системи зору людини (HVS). Візуальні атаки — це простий метод стеганоаналізу, який може зламати стеганографію за допомогою HVS. Стегокаркас і рамку кришки порівнюються неозброєним оком, щоб перевірити наявність видимих змін. Хоча візуальні атаки легко реалізувати, вони не є надійними. Іншими недоліками використання візуальної атаки є не тільки надійність, але й автоматизація та вимога до експертів для проведення тестування.

Характеристики та особливості обкладинки змінюються після введення секретної інформації. Для визначення наявності секретної інформації враховуються деякі характеристики, особливості та інші структурні компоненти стеговідео. Цей тип атаки стегааналізу називається структурною атакою. Одним із прикладів є порівняння розміру файлу. Після вставлення розмір файлу обкладинки відео може змінюватися. Подібно до візуальних атак, структурні атаки не є надійними, тому потрібні експерти в цій галузі. Не всі підроблені контейнери зазнають структурних змін і можуть уникнути методів структурної атаки.

### 1.3.2 Статистичний стегоаналіз

Статистичний стеганаліз використовує значення пікселів зображення та аналізує їх для виявлення конфіденційного вмісту. Статистичний стеганаліз є провідним у порівнянні з сигнатурним стеганалізом. Статистичні методи використовують знання значень пікселів зображення та математичні моделі для виявлення та відновлення секретної інформації. Статистичний стеганаліз можна згрупувати за аналізом гістограми, хі-квадрат-атаками, RS-стеганалізом, вбудовуванням LSB, аналізом піксельних пар із збігом LSB, аналізом бітової площини, стисненням JPEG і стеганалізом домену перетворення.

Аналіз гістограми стеганалізу гістограми контейнеру та стегограми для виявлення присутності секретної інформації. Гістограма — це графічне представлення значень пікселів зображення на основі розподілу. Коли контейнер змінюється для вбудовування секретної інформації, це впливає на гістограму. Вбудовування секретної інформації може бути невидимим для людського ока, але коли гістограма будується та порівнюється з оригінальним контейнером, можна виявити навіть найменшу маніпуляцію.

Критерій хі-квадрат — це поширений метод стеганалізу, який використовується для виявлення присутності секретної інформації. Цей тест працює шляхом спостереження за подібністю між подією в реальному часі та очікуваним результатом. Він використовує розподіл частот для визначення випадковості у контейнері. Більш низькі значення тесту вказують на більш високий ступінь випадковості, підтверджуючи наявність секретного повідомлення. Вищі значення означають нижчий ступінь випадковості та доводять відсутність фальсифікацій у відео.

RS стегоаналіз є ще одним потужним інструментом Аналіз RS використовується для виявлення секретної інформації, яка була вбудована за допомогою методів на основі LSB. Він порівнює значення пікселів зображення в просторовій області. Вибір пар пікселів залежить від методу, іноді для порівняння вибираються сусідні пікселі, а в інших випадках – пікселі з різних блоків. Ці групи пікселів називаються сингулярними групами (S) і регулярними групами (R). Наявність секрету визначається шляхом групування пікселів на основі розподілу частот і аналізу LSB стего та відео обкладинки. LSB перевертаються та рандомізуються для виявлення секретного повідомлення. Аналіз RS має кращу надійність порівняно з тестами хі-квадрат.

Методи стеганографічного аналізу LSB вбудовування та узгодження LSB базуються на принципі роботи методу стеганографії LSB. Ці методи стеганографії популярні, оскільки використання методів стеганографії LSB є широким порівняно з іншими методами стеганографії. Стеганаліз домену трансформації перетворює зображення в частотну область з величиною та фазою. Амплітуда представляє

значення підрахунку частоти зображення, а фаза представляє напрямок відновлення початкової форми зображення. Зазвичай використовуваними підходами для області перетворення є вейвлет, Фур'є та косинус.

### 1.3.3 Стегоаналіз на основі ознак

Риси є важливою частиною образу. Методи стегоаналізу на основі ознак виділяють ознаки із зображень і аналізують їх, щоб виявити наявність секретної інформації. Ці функції можна додатково використовувати для навчання класифікатора автоматичному виявленню секретної інформації за допомогою алгоритмів машинного навчання. Методи стеганографії з різницею значень пікселів (PVD) приховують більше бітів секретної інформації в гладкіших областях зображення обкладинки, ніж у складних областях. Аналіз гістограми стеганографії PVD виявив розподіл Лапласа. Для виявлення присутності секретної інформації використовується метод стегоаналізу на основі ознак. Оскільки PVD має розподіл Лапласа, очікуваний розподіл частоти зображення отримується за допомогою будь-якого тесту на випадковість. Очікувані значення порівнюються із спостережуваними значеннями та обчислюється ступінь подібності. Якщо подібність нижче певного порогу, то зображення не було підроблено, інакше воно має деяку вбудовану інформацію.

## 1.4 Висновки. Постановка задачі

У цьому розділі булу розглянуті основні стеганографічного методи для таких типів контейнерів як: цифрове зображення, текст, відео та аудіо. Для контейнеру цифрового зображення було розглянуто методи стеганографії як: LSB, JPEG стискання, Patchwork та розширення спектру; для текстового контейнера – Зсув рядка, зсув слова, синтаксичний метод, білий стенограф та спам-текст; для відео – DWT, WST, DCT та вектори руху; для аудіо – низькорозрядне кодування, приховування у відлунні, у інтервалах тиші та вставка тону. Також було проаналізовано такі методи стегоаналізу як: сигнатурний, статистичний та стегоаналіз на основі ознак.

У спеціальній частині необхідно: розглянути відкрите програмне забезпечення цифрової стеганографії та стегоаналізу. Наступним кроком є дослідження особливостей відкритих Python бібліотек стеганографії та стегоаналізу. Кінцевим кроком є дослідження сумісності відкритих Python бібліотек та розглянутого програмного забезпечення.

# РОЗДІЛ 2. СПЕЦІАЛЬНА ЧАСТИНА

## Вступ

На сьогодні існує безліч інструментів, які реалізують різні методи стеганографічного захисту інформації - для зображень, аудіо та відео. Є як і безкоштовні інструмент, так і платні. У даній частині кваліфікаційній роботі розглядаються лише безкоштовні інструменти захисту. Для деяких з них код програми є у відкритому доступі.

## 2.1 Дослідження програмного забезпечення стеганографії та стегоаналізу

### 2.1.1 StegoShare

StegoShare - безкоштовна програма з відкритим кодом для стеганографічного захисту інформації, що використовує мережу P2P для розповсюдження закодованих зображень, написаний мовою програмування Java. За допомогою неї можна заховати файл розміром до 2 ГБ в декілька зображень. Програма підтримує майже всі графічні формати. Це робить програму ідеальним інструментом для приховування цензурованих файлів, розповсюдження яких може бути незаконним у певній країні, у легальні фото, з метою подальшого розповсюдження на торрент-платформах. В основі роботи програми лежить LSB метод. На рисунку 2.1 зображено головне меню програми

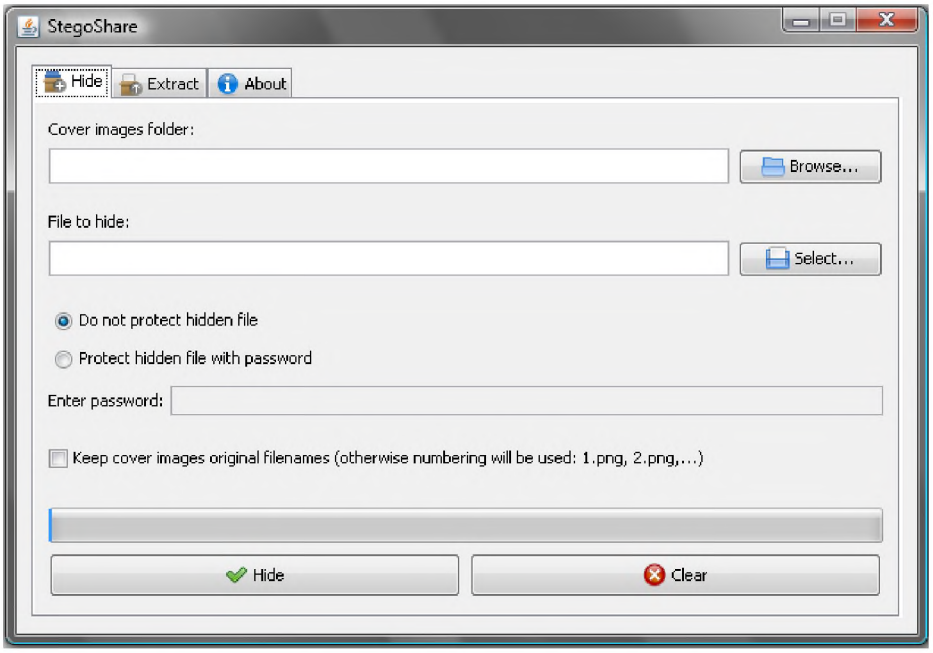


Рис. 2.1 – Головне меню програми StegoShare

### 2.1.2 Steghide

Steghide є програмою для стеганографії, яка дозволяє приховувати дані у різних типах зображень та аудіо файлів (рис. 2.2). При цьому кольори та частоти дискретизації не змінюються, що робить вбудовування стійким до статистичного аналізу.

Програма включає в себе:

- Стиснення даних, що вбудовуються;
- Шифрування даних, що вбудовуються;
- Автоматична перевірка цілісності за допомогою контрольної суми;
- Підтримка форматів файлів-контейнерів JPEG, BMP, WAV і AU;
- Немає обмежень на формат секретних даних

Steghide використовує графо-теоретичний підхід до стеганографії. Відбувається це наступним чином:

1. секретні дані стискаються та шифруються;
2. створюється послідовність позицій пікселів у контейнері на основі псевдовипадкового генератора чисел, ініціалізованого парольною фразою (секретні дані будуть вбудовані в пікселі на цих позиціях);
3. Серед цих позицій відсортовуються ті, які не потрібно змінювати (оскільки вони вже випадково містять правильне значення);
4. графо-теоретичний алгоритм зіставлення знаходить пари позицій таким чином, що обмін їхніми значеннями призводить до вбудовування відповідної частини секретних даних;
5. якщо алгоритм не може знайти більше таких пар, то всі обміни фактично виконуються
6. Пікселі на решти позицій (позиції, які не є частиною такої пари) також модифікуються для розміщення вбудованих даних (але це робиться шляхом перезапису, а не обміну їх з іншими пікселями);
7. той факт, що (більшість) вбудовування здійснюється шляхом обміну значеннями пікселів, означає, що статистика (тобто кількість разів, коли колір зустрічається на зображенні) не змінюється.

Алгоритмом шифрування за замовчуванням є Rijndael з розміром ключа 128 біт (що є AES – Advanced Encryption Standard) у режимі ланцюгового шифрування блоків. Існує можливість вільно вибрати іншу комбінацію алгоритму/режиму (інформацію про всі можливі алгоритми та режими можна отримати за допомогою команди encinfo). Контрольна сума обчислюється за допомогою алгоритму CRC32.

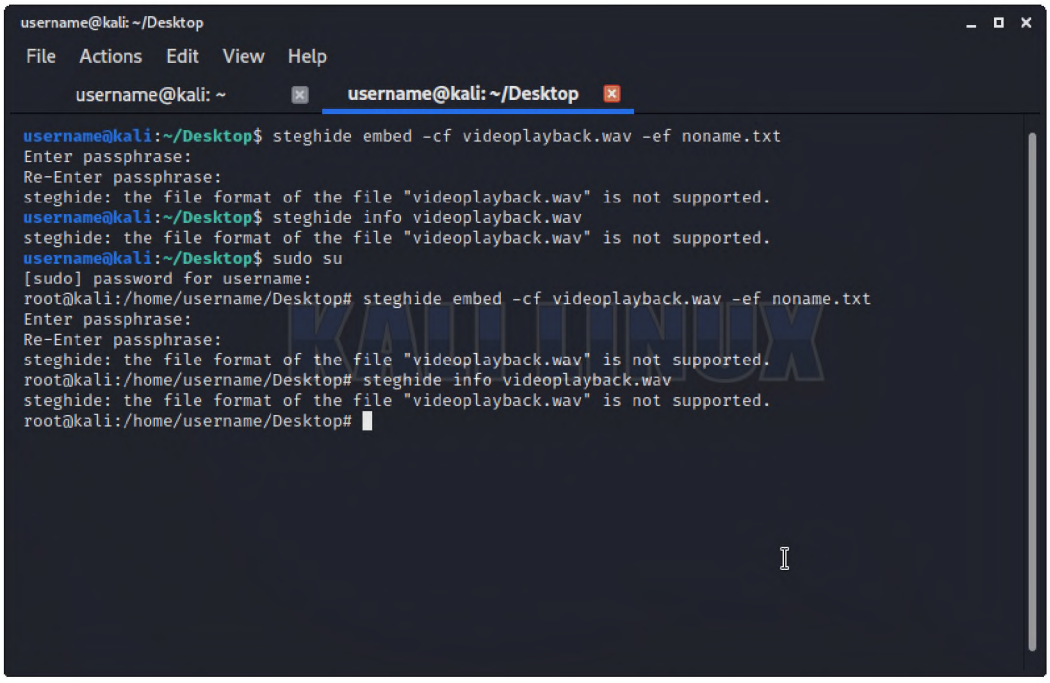


Рис. 2.2 – приклад використання програми steghide через термінал Kali Linux

### 2.1.3 S-Tools

S-Tools (рис. 2.3) - це простий інструмент, який можна використовувати для приховування текстових файлів у зображеннях [9]. Потрібно запустити .exe-файл і перетягнути BMP-файл у вікно. Потім перетягнути текстовий файл на зображення і вибрати алгоритм шифрування для вбудовування файлу. Доступні наступні варіанти: IDEA, DES, потрійний DES тощо. Також можна вибрати ключову фразу, яка буде використана для відкриття прихованих файлів із зображень. Той самий процес, щоб відкрити повідомлення, яке було приховано у файлі.

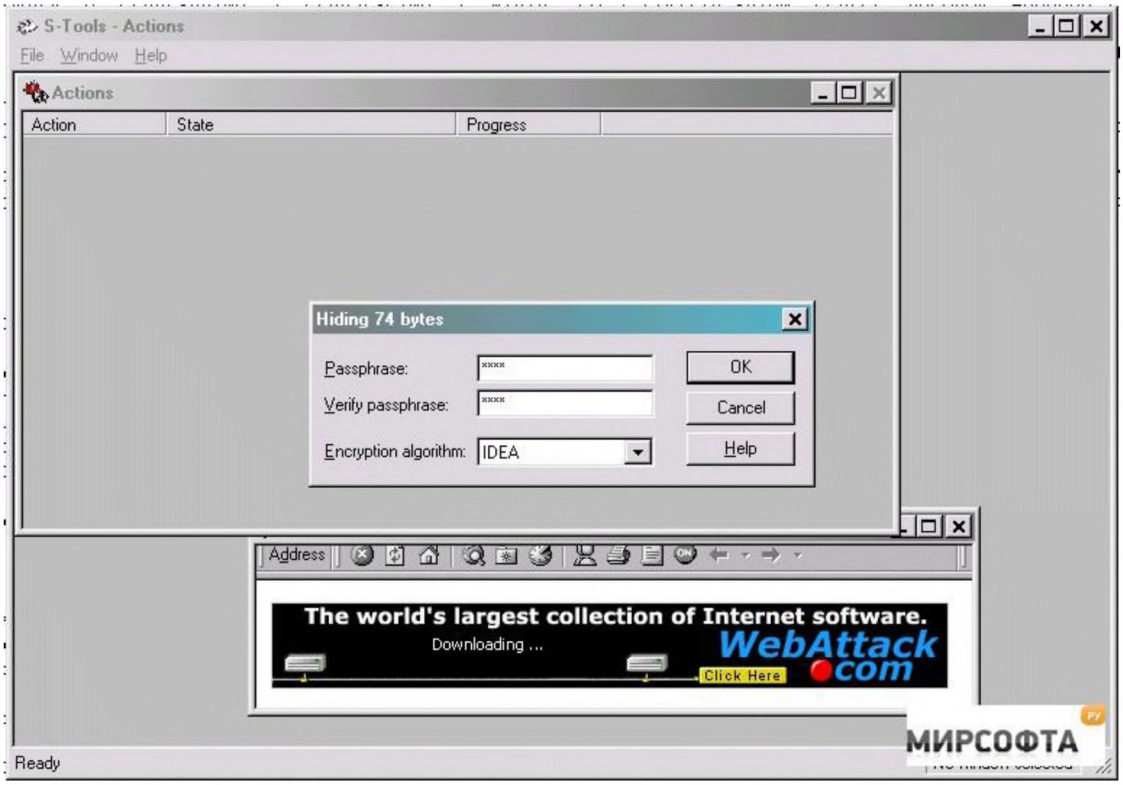


Рис. 2.3 – головний екран програми S-Tools з демонстрацією шифрування повідомлення

### 2.1.4 OpenStego

OpenStego [8]- безкоштовна програма для приховування даних контейнері зображень. Підтримує технологію цифрового водяного знаку (ЦВЗ). Написана мовою програмування Java. Програма може працювати у двох режимах: графічному (рис. 2.4) та у терміналі (рис. 2.5).

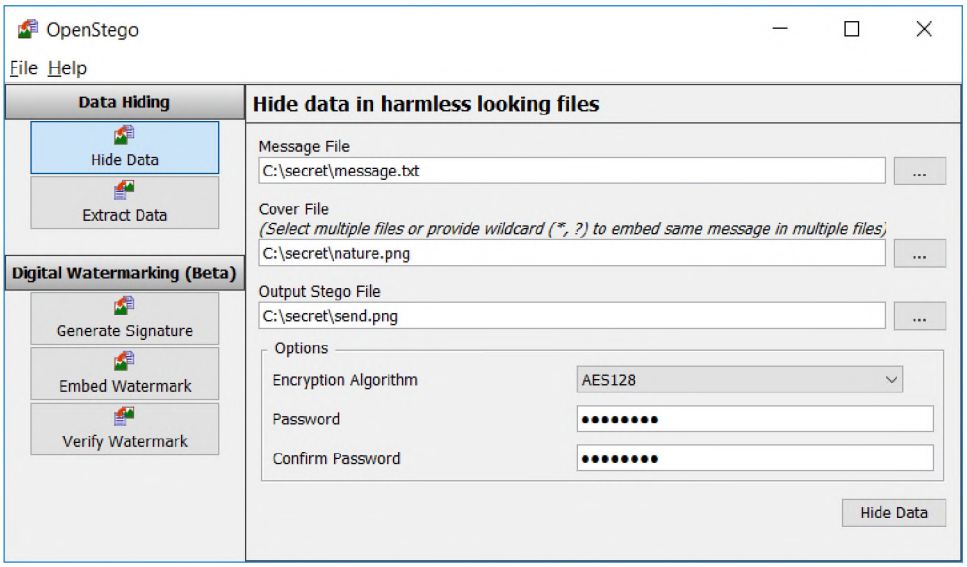


Рис. 2.4 – головне меню програми OpenStego у графічному режимі

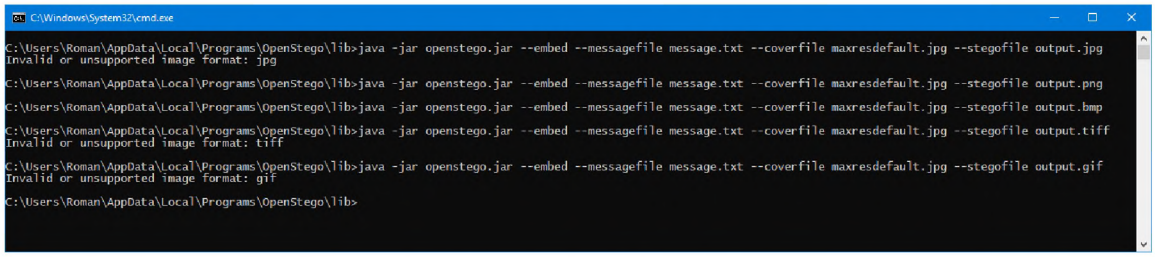


Рис. 2.5 – приклад використання програми OpenStego у термінальному режимі

## 2.2 Дослідження Python бібліотек стеганографічного захисту інформації

### 2.2.1 stegolsb

Дана бібліотека є найбільшою серед існуючих бібліотек на Python [10]. Вона реалізує LSB для зображень, аудіо даних та інших потоків даних. Складається із наступних модулів: Byte Sequence Manipulation, WavSteg, LSBSteg та StegDetect. Розглянемо кожен окремо.

Byte Sequence Manipulation відповідає за приховування довільної бітової послідовності. У його основі лежить алгоритм швидкого переміщення байтів цільового повідомлення безпосередньо у молодших бітах послідовності байтів контейнеру.

WavSteg використовує метод найменш значущого біту, щоб приховувати текстову інформацію в .wav файли. Для кожного байту у аудіофайлі перезаписуються молодші біти на дані з нашого контейнеру.

LSBSteg використовує стеганографію найменших значущих бітів, щоб приховати файл у кольоровій інформації RGB-зображення (.bmp або .png). Для кожного колірного каналу (наприклад, R, G і B) в кожному пікселі зображення перезаписуються молодші біти значення кольору даними з файлу. Щоб полегшити відновлення цих даних, також приховується розмір вхідного файлу в перших кількох колірних каналах зображення. Перед тим, як приховувати дані у зображені, необхідно дізнатися, який об’єм даних можна приховати за допомогою вбудованих методів у бібліотеку.

### 2.2.2 Exstego

Exstego - повноцінний додаток написаний мовою програмування Python для стеганографії [11]. Особоливості:

1. Працює у двох режимах:
    a. Scatter – кодує цільове повідомлення з рівномірним розподілом по контейнеру;
    b. Sequential – кодує, декодує або знищує цільове повідомлення починаючи з першого пікселя.
2. Реалізує три методи кодування:
    a. LSB – стандартний алгоритм роботи заміни менш значущих біт
    b. METADATA – приховування повідомлення у метаданих (заголовках) файлу
    c. BPCS (Bit Plane Complexity Segmentation)

Розглянемо BPCS більш детально.

Цифрова стеганографія може дуже надійно приховувати конфіденційні дані, вбудовуючи їх у деякі дані на носії, які називаються "контейнерами". Дані контейнеру також називають «даними носія», «обкладинки» або «фіктивними даними». У BPCS (абревіатура від *Bit-Plane Complexity Segmentation*) стеганографії для даних контейнерц здебільшого використовуються повнокольорові зображення (тобто 24-розрядні кольорові зображення). Операція вбудовування на практиці полягає в заміні «складних областей» на бітових площинах зображення на конфіденційні дані. У порівнянні з простою стеганографією на основі зображень, яка використовує лише найменш важливий біт даних, і тому (для 24-бітового кольорового зображення) може вбудовувати дані, еквівалентні 1/8 від загального розміру, BPCS-стеганографія використовує декілька бітових площин, і тому може вбудовувати набагато більшу кількість даних, хоча це залежить від конкретного зображення. Для «нормального» зображення приблизно 50% даних можуть бути замінені секретними даними до того, як деградація зображення стане очевидною.

Зорова система людини має таку особливу властивість, що занадто складний візуальний шаблон може не сприйматися як "інформативний щодо форми". Наприклад, на дуже пласкому пляжному березі кожен квадратний фут виглядає однаково - це просто піщана поверхня, ніякої форми не спостерігається. Однак, якщо придивитися уважніше, дві ділянки, що виглядають однаково, абсолютно

відрізняються за формою піщинок. BPCS-стеганографія використовує цю властивість. Вона замінює складні області на бітових площинах зображення іншими складними шаблонами даних (тобто фрагментами секретних файлів). Ця операція заміни називається «вбудовуванням». Ніхто не може побачити різниці між двома зображеннями до і після операції вбудовування.

### 2.2.3 LSB-Steganography

Модуль LSBSteg [13] заснований на OpenCV, щоб приховати дані в зображеннях. Він використовує перший біт кожного пікселя та кожен колір зображення. Якщо використано кожен перший біт, модуль починає використовувати другий біт, отже, чим більші дані, тим більше змінюється зображення. Програма може приховати всі дані, якщо контейнер має достатній розім. Основні функції:

1. encode_text: отримує рядок з текстовим повідомленням, а бібліотека його приховує;
2. encode_image: бібліотека отримує зображення OpenCV, і метод повторює кожен піксель, щоб приховати їх. Хороша практика полягає в тому, щоб мати контейнер у 8 разів більший за цільове зображення, яке потрібно приховати (щоб кожен піксель містився лише в першому біті).
3. encode_binary: бібліотека отримує двійковий файл для приховування. Працює за аналогічним принципом як і encode_image. Цей метод може приховати будь-який файл.

## 2.3 Дослідження Python бібліотек для стегоаналізу

### 2.3.1 Exstego

Програма Exstego має потужний модуль для стегоаналізу. Він має декілька методів стегоаналізу (атак):

1. аналітичний: Для нього необхідно мати оригінальне зображення та зображення з прихованим повідомленням. Має можливість розкладання бітів на так звані «шари» та порівняння на основі гістограм.

2. «стегоорієнтовний»: накладання та розбивання «шарів» бітів, видаляння або отримання метаданих цільового повідомлення (у випадку приховування зображень або аудіо файлів)

2.3.2 StegExpose

StegExpose [12] - це інструмент стегоаналізу, що спеціалізується на виявленні LSB (найменш значущого біта) стеганографії в зображеннях без стиснення, таких як PNG і BMP. Він має інтерфейс командного рядка, надаючи при цьому можливості звітування та налаштування, зрозумілі для експертів, які не є фахівцями з криміналістики. Алгоритм оцінки StegExpose заснований на ретельно протестованій комбінації вже існуючих методів піксельного стегоаналізу: вибіркові пари Думітреску, RS-аналіз, атака “Хі-квадрат” і первинні множини Думітреску. На додаток до виявлення стеганографії, StegExpose також має функцію кількісного стегоаналізу (визначення довжини прихованого повідомлення).

2.3.3 Модуль StegDetect бібілотеки Stego-LSB

Модуль реалізує метод розпізнавання стеганографії у цифрових зображеннях. Принцип дії наступний: підсумовує молодші n біт RGB кольорів для кожного пікселя і нормалізує результат в діапазоні 0–255. Це значення застосовується до кожного колірного каналу пікселя. Де n — кількість молодших бітів, які потрібно відобразити. Цей алгоритм працюватиме виключно з зображеннями без стискання, оскільки біти LSB можуть бути змінені на етапі стискання.

2.4 Дослідження сумісності

Для виконання дослідницької роботи були обрані наступні бібліотеки: stegolsb, LSB-Steganography

По горизонталі розташовані бібліотеки та програми, що реалізують стеганографічні методи захисту інформації, а по вертикалі – бібліотеки для стегоаналізу.

### 2.4.1 Stego-lsb

Для початку оберемо бібліотеку stegolsb. Відповідно до офіційної інструкції, її необхідно встановити за допомогою пакетного менеджеру pip (рис. 2.6).

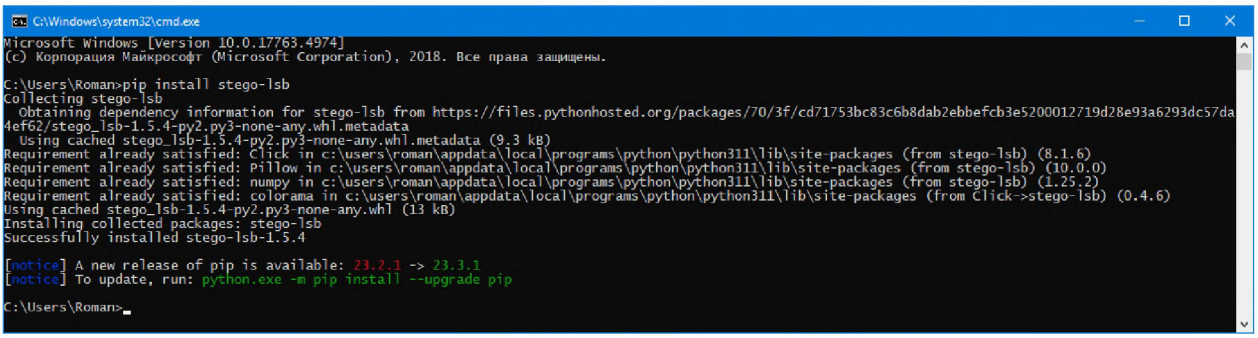


Рис. 2.6 – Процес встановлення бібліотеки stegolsb

Наступним кроком буде підключення бібліотеки та звернення до функції, яка реалізує приховування вхідного текстового файлу з секретним повідомленням у вхідному зображенні (рис.2.7 ). Для тесту було обрано наступне зображення (рис. 2.8).

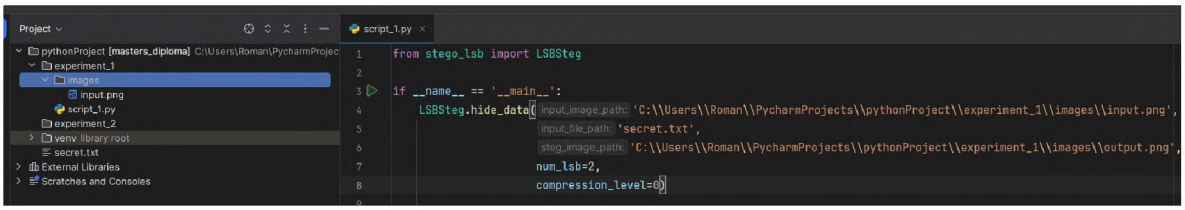


Рис. 2.7 – Демонстрація підключення бібліотеки та звернення до функції приховування

Рис. 2.1 – Тестове зображення

У якості повідомлення буде приховано текстовий файл зі словами із пісні групи «Kiss – I Was Made For Lovin' You» (рис. 2.9)

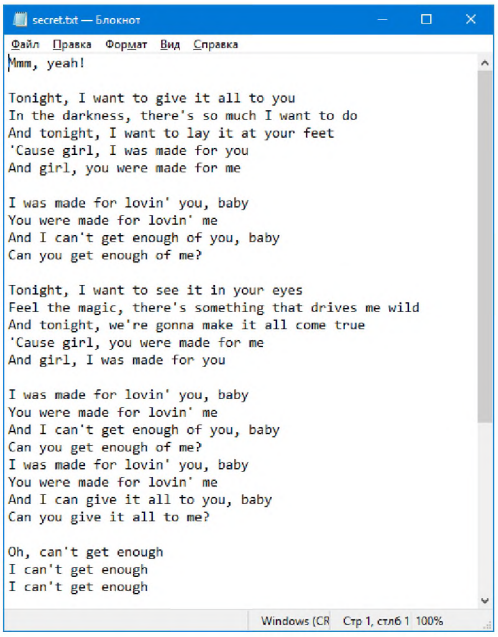


Рис. 2.2 – Цільове повідомлення, що буде приховано

Після виконання скрипту було отримано нове зображення (рис. 2.10)

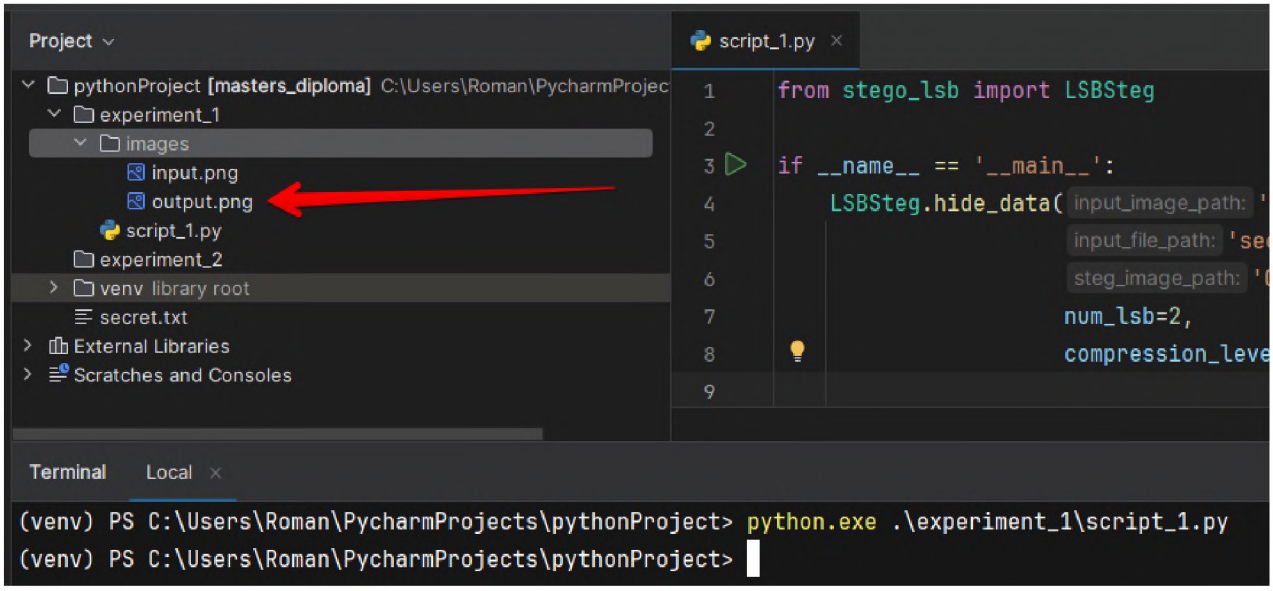


Рис. 2.3 – Результатом виконання роботи бібліотеки став новий файл

Проаналізуємо результати. По-перше, розмір нового файлу більший у 2,23 рази, ніж оригінал (рис. 2.11). По-друге, візуально зображення мало чим відрізняються (рис. 2.12).

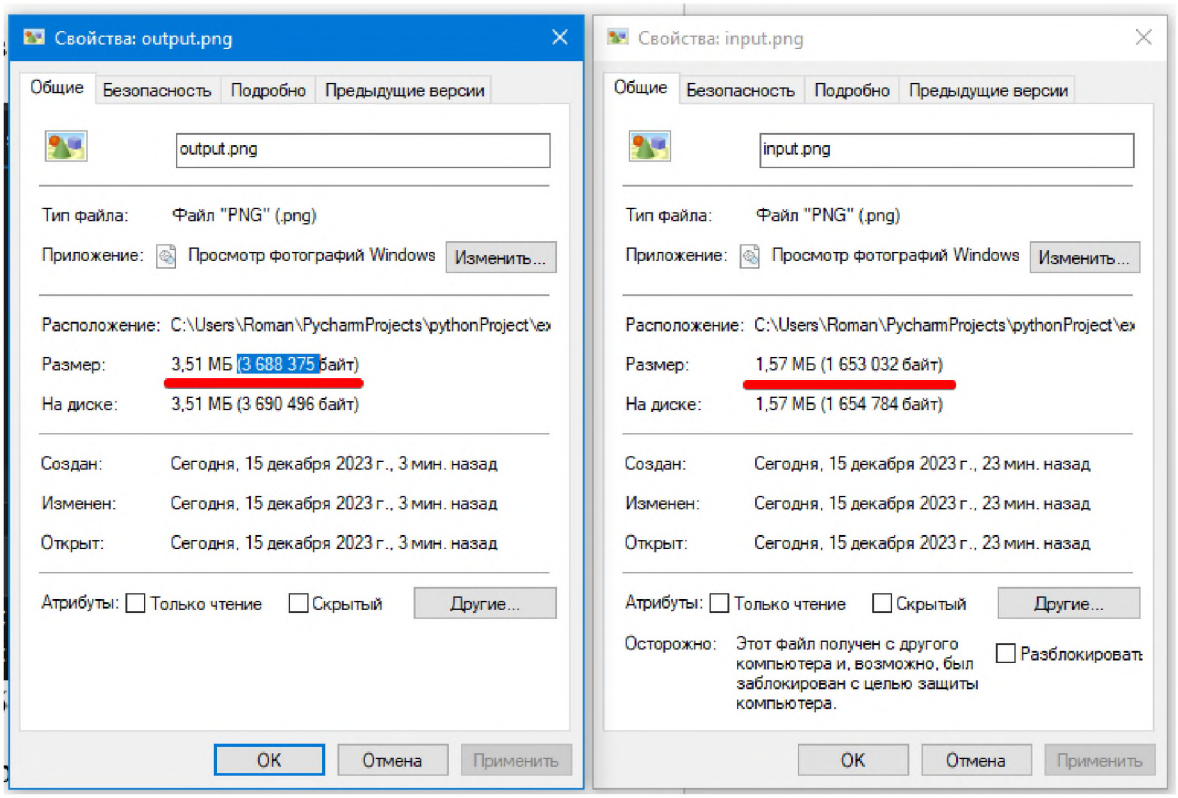


Рис. 2.4 – Порівняння розмірів вхідного та вихідного файлів

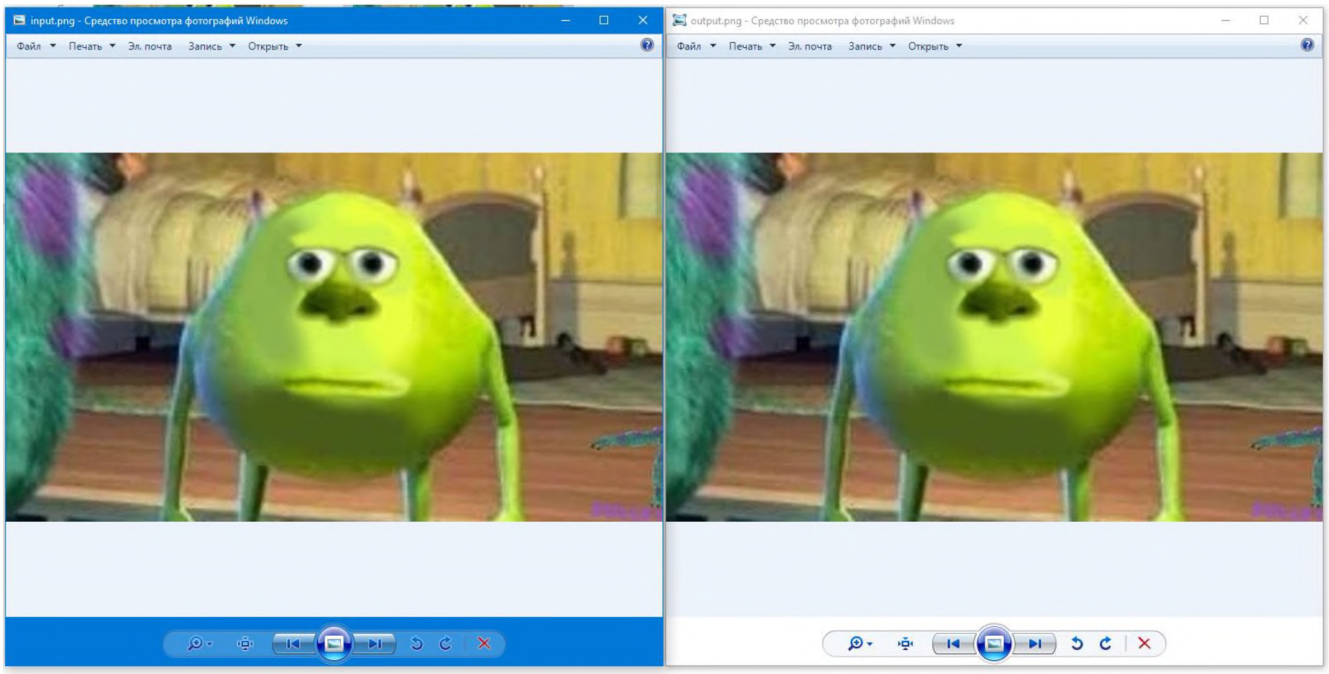


Рис. 2.5 – Візуальне порівняння оригіналу та вихідного зображень

Проведемо стегоаналіз за допомогою обраних бібліотек. Першою бібліотекою буде LSB-Steganography. З її допогою будемо отримувати бінарний файл. Результатом стегоаналізу має бути декодований прихований текстовий файл (рис. 2.13).

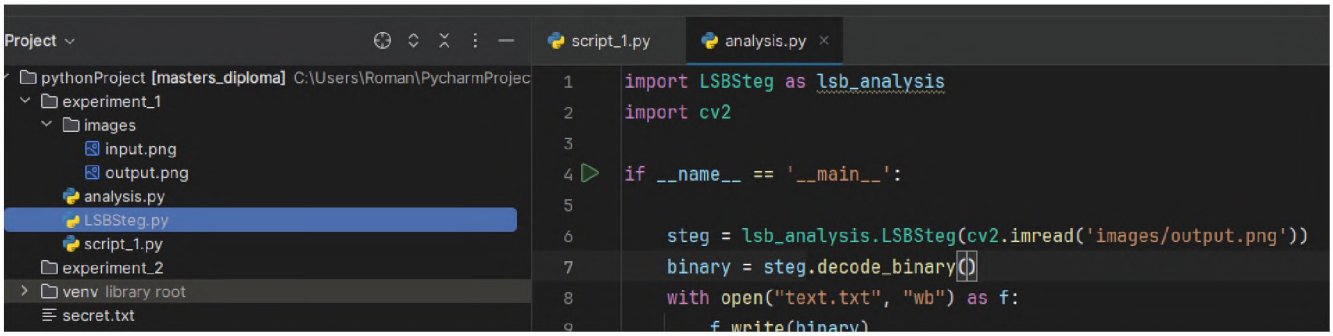


Рис. 2.6 – Приклад коду для проведення стегоаналізу за допомогою бібліотеки LSB-Steganography

Роботу стегоаналізу довелося призупинити з причини надто довгої та безрезультатної роботи (рис. 2.14).

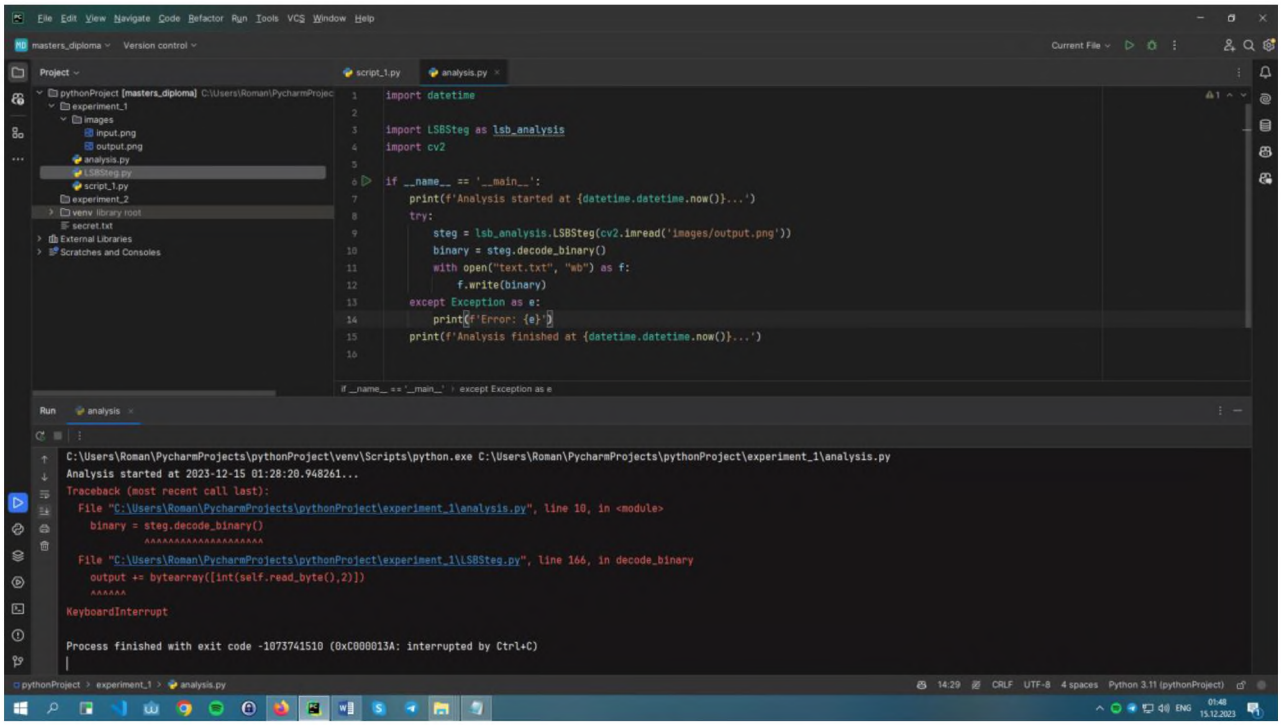


Рис. 2.7 – Помилка роботи стегоаналізу

Наступним інструментом для стегоаналізу стане модуль для декодування бібліотеки stego-lsb. Результатом стегоаналізу має бути вихідний тестовий файл (рис. 2.15, рис. 2.16)

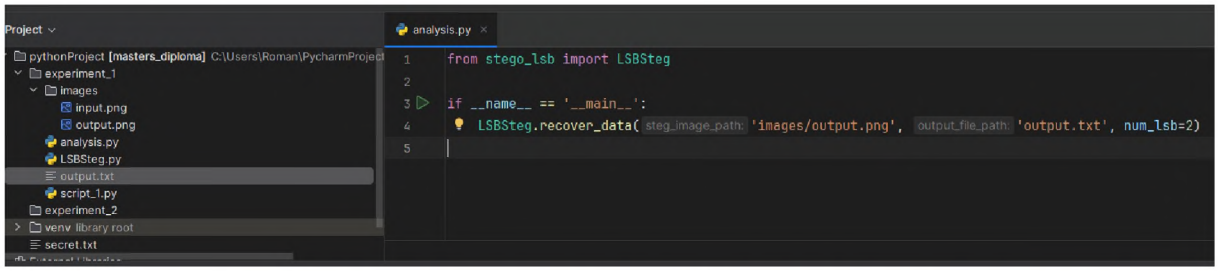


Рис. 2.8 - Приклад коду для проведення стегоаналізу за допомогою бібліотеки stego-lsb

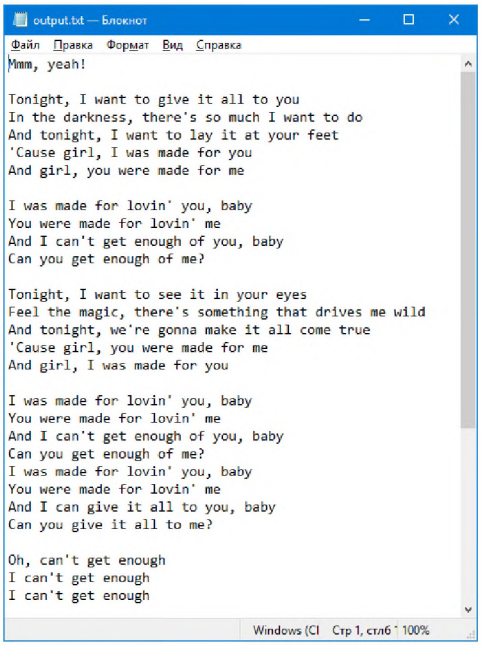


Рис. 2.9 – Результат роботи стегоаналізу

### 2.4.2 LSB-Steganography

Наступною бібліотекою для дослідження стає LSB-Steganography. Відповідно до офіційної документації, необхідно завантажити головний файл (LSBSteg.py) з репозиторію та покласти його у проект (рис, 2.17).

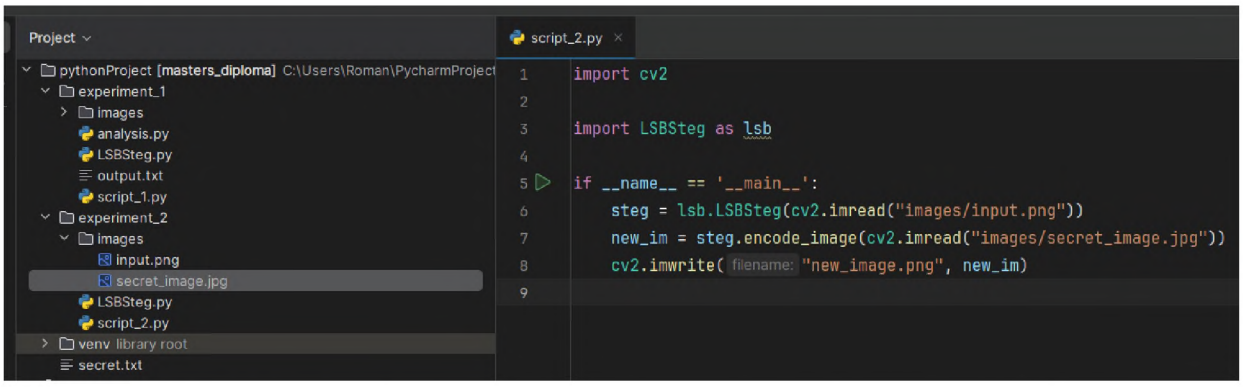


Рис. 2.17 – Приклад використання коду програми LSB-Steganography

При спробі приховати у вхідне зображення тестове зображення, отримуємо помилку бібліотеки. Причиною цьому стала несумісність даної бібліотеки та бібліотеки, яку використовую LSB-Steganography (рис. 2.18).

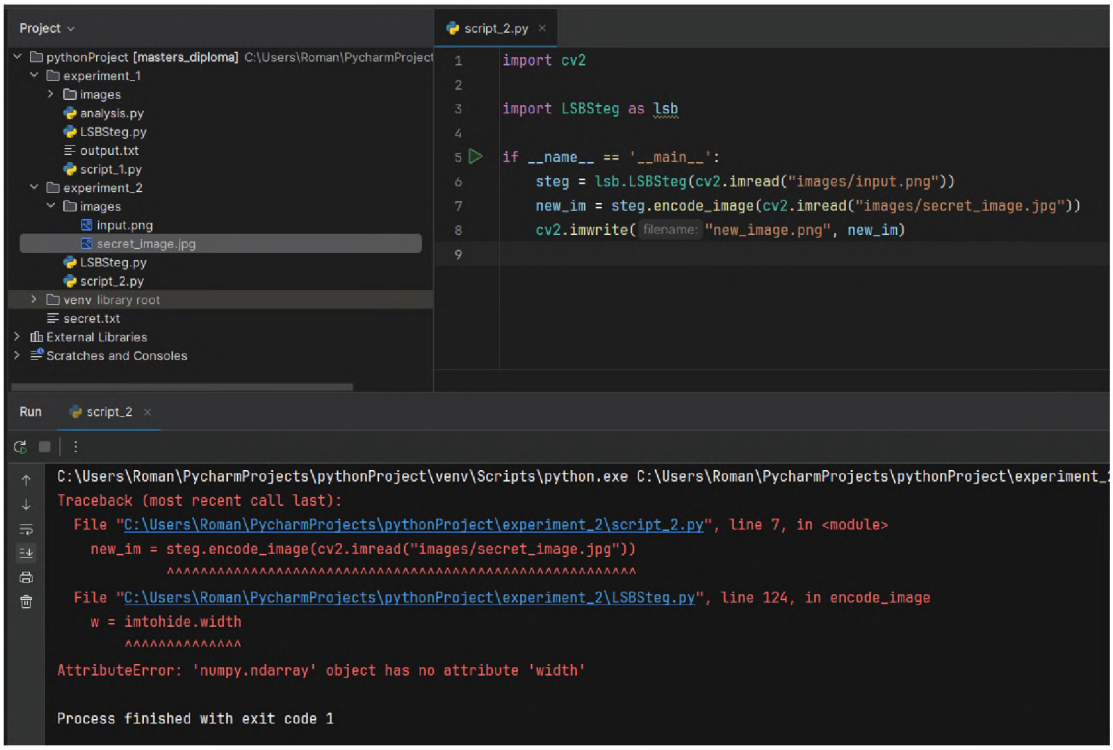


Рис. 2.18 – Демонстрація помилки під час роботи

Висновки

У даному розділі було проаналізовано які існують програмні засоби стеганографічного захисту інформації (стороні програми та бібліотеки написані іншими мовами), розглянуті бібліотеки для стегоаналізу та проведено дослідження використання Python бібліотек у процесі використання цифрової стеганографії.

Відповідно до проведеного дослідження можна зробити висновок, що використання розглянутих Python бібліотек не є доцільним, оскільки вони не сумісні один з одним через різну реалізацію. Особливу увагу варто приділяти тим бібліотекам, які реалізуються pure методом (без використання сторонніх бібліотек). Розглянуті бібліотеки можна використовувати у закритих системах – у межах однієї програми з використанням однієї бібліотеки, що реалізує обидва методи – приховування та декодування.

# РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ

## Вступ

У цьому розділі кваліфікаційної роботи метою є техніко-економічне обґрунтування доцільності дослідження Python бібліотек у процесі використання цифрової стеганографії. Порахувати капітальні, експлуатаційні витрати, дослыдити економічний ефект та проаналізувати показники економічної ефективності дослідження Python бібліотек.

## 3.1 Визначення капітальних витрат на дослідження Python бібліотек

Капітальні витрати – це кошти, призначені для створення і придбання основних фондів і нематеріальних активів, що підлягають амортизації.

### 3.1.1 Визначення капітальних витрат на дослідження Python бібліотек

Трудомісткість розробки КСЗІ визначається тривалістю кожної робочої операції, починаючи з складання технічного завдання і закінчуючи оформленням документації (за умови роботи одного спеціаліста з інформаційної безпеки):

$$t = t_{\text{тз}} + t_{\text{в}} + t_{\text{опр}} + t_{\text{д}}, \text{ годин, де}$$

$t_{\text{тз}}$- тривалість складання ТЗ на розробку ПЗ;

$t_{\text{в}}$- тривалість вивчення ТЗ, літературних джерел за темою тощо;

$t_{\text{опр}}$- тривалість опрацювання алгоритмів стеганографії на ПК;

$t_{\text{д}}$- тривалість підготовки технічної документації на ПЗ.

$$t = 24 + 48 + 12 + 6 = 90$$

### 3.1.2 Розрахунок витрат на створення програмного продукту

Витрати на створення програмного продукту $К_{\text{пз}}$ складаються з витрат на заробітну плату виконавця ПЗ $З_{\text{зп}}$ і вартості витрат машинного часу, що необхідних для опрацювання програми на ПК $З_{\text{мч}}$

$$К_{\text{пз}} = З_{\text{зп}} + З_{\text{мч}} = 12600 + 36{,}48 = 12636{,}48 \text{ грн}$$

Заробітна плата виконавця враховує основну і додаткову заробітну плату, а також відрахування на соціальне потреби (пенсійне страхування, страхування на випадок безробіття, соціальне страхування тощо) і визначається за формулою:

$$З_{\text{зп}} = t \cdot З_{\text{пр}} \text{ грн, де}$$

$t$ - загальна тривалість створення ПЗ, годин;

$З_{пр}$ - середньогодинна заробітна плата програміста з нарахуваннями, грн/годину

$$З_{зп} = 90 \cdot 140 = 12600 \text{ грн}$$

Вартість машинного часу для налагодження програми на ПК визначається за формулою:

$$З_{мч} = t_{опр} \cdot С_{мч} + t_{д} \text{ грн, де}$$

$t_{опр}$- тривалість опрацювання алгоритмів стеганографії на ПК, година;

$t_{д}$- тривалість підготовки технічної документації на ПЗ, година;

$С_{мч}$ - вартість 1 години машинного часу на ПК, грн/годин

У свою чергу вартість 1 години машинного часу на ПК визначається за формулою:

$$С_{мч} = P \cdot t_{нал} \cdot C_e + \frac{Ф_{зал} \cdot Н_а}{F_P} + \frac{К_{лпз} \cdot Н_{апз}}{F_P} \text{ грн/гож, де}$$

$P$ - встановлена потужність ПК, кВт

$t_{нал}$ - кількість ПК, шт

$C_e$ - тариф на електричну енергію, грн/кВ*тодина

$Ф_{зал}$ - залишкова вартість ПК на поточний рік, грн

$Н_а$ - річна норма амортизації на ПК, частки одиниці

$К_{лпз}$ - вартість ліцензійного програмного забезпечення, грн

$Н_{апз}$ - річна норма амортизації на ліцензійне програмне забезпечення, частки одиниці

$F_P$ - річний фонд робочого часу (за 40-годинного робочого тижня Fp=1920)

$$С_{мч} = 0{,}15 \cdot 1 \cdot 2{,}64 + \frac{13750 \cdot 0{,}3}{1920} + \frac{0 \cdot 0}{1920} = 0{,}396 + 2{,}1484375 = 2{,}54 \text{ грн}$$

Вартість ноутбуку 22000 грн, строк служби 36 місяців

Накопичена амортизація: $\frac{22000 \cdot 36}{8 \cdot 12} = 8250$ грн

Залишкова вартість: $22000 - 8250 = 13750$ грн

Відповідно до нарахованої вартості машинного часу

$$З_{мч} = 12 \cdot 2{,}54 + 6 = 36{,}48$$

Таким чином, капітальні (фіксовані) витрати на дослідження Python бібліотек складають:

$$K = К_{пр} + К_{зпз} + К_{пз} + К_{аз} + К_{навч} + К_{н}, \text{ де}$$

$К_{пр}$ – вартість розробки проекту інформаційної безпеки та залучення для цього зовнішніх консультантів, тис. грн;

$К_{зпз}$ – вартість закупівель ліцензійного основного й додаткового програмного забезпечення (ПЗ), тис. грн;

$К_{пз}$ – вартість розробки програмного продукту, тис. грн;

$К_{аз}$ – вартість закупівлі апаратного забезпечення та допоміжних матеріалів, тис. гри;

$К_{навч}$ – витрати на навчання технічних фахівців і обслуговуючого персоналу, тис. грн;

$К_{н}$ – витрати на встановлення обладнання та налагодження системи інформаційної безпеки, тис. Грн.

$$K = 0 + 0 + 12636{,}48 + 0 + 0 + 0 = 12636{,}48 \text{ грн}$$

Таким чином, капітальні витрати К=12636,48 грн.

## 3.2 Визначення експлуатаційних витрат на дослідження Python бібліотек

Експлуатаційні витрати - це поточні на експлуатацію та обслуговування об'єкта проектування за визначений період (наприклад, рік), що виражені у грошовій формі.

Відповідно до теми кваліфікаційної роботи, дослідження є одноразовим, експлуатаційних витрат не потребує у разі інтегрування результатів дослідження у систему розробки.

## 3.3 Оцінка можливого збитку

Кінцевим результатом впровадження й проведення заходів щодо забезпечення інформаційної безпеки є величина відвернених втрат, що розраховується, виходячи з імовірності виникнення інциденту інформаційної безпеки й можливих економічних втрат від нього. Фактично, ця величина відображає ту частину прибутку, що могла бути втрачена.

3.3.1 Оцінка величини збитку

Для розрахунку вартості такого збитку можна застосувати наступну спрощену модель оцінки. Необхідні вихідні дані для розрахунку:

$t_{\text{п}}$ — час простою вузла або сегмента корпоративної мережі внаслідок атаки, 2 годин;

$t_{\text{в}}$ — час відновлення після атаки персоналом, що обслуговує корпоративну мережу, 1 година;

$t_{\text{ви}}$ — час повторного введення загубленої інформації співробітниками атакованного вузла або сегмента корпоративної мережі, 5 годин;

$З_{\text{о}}$ — заробітна плата обслуговуючого персоналу (адміністраторів та ін.), 0 грн на місяць;

$З_{\text{с}}$ - заробітна плата співробітників атакованого вузла або сегмента корпоративної мережі, 15000 грн на місяць;

$Ч_{\text{о}}$ - чисельність обслуговуючого персоналу (адміністраторів та ін.), 0 осіб;

$Ч_{\text{с}}$- чисельність співробітників атакованого вузла або сегмента корпоративної мережі, 1 осіб.;

О - обсяг прибутків атакованого вузла або сегмента корпоративної мережі, 500 тис. грн. у рік;

$П_{\text{зч}}$ - вартість заміни устаткування або запасних частин, 3000 грн;

I – число атакованих вузлів або сегментів корпоративної мережі; 1

N - середнє число атак на рік. 5

Упущена вигода від простою атакованого вузла або сегмента корпоративної мережі становить:

$$U = П_{\text{п}} + П_{\text{в}} + V, \text{ де}$$

$П_{\text{п}}$ - оплачувані втрати робочого часу та простої співробітників атакованого вузла або сегмента корпоративної мережі, грн;

$П_{\text{в}}$ - вартість відновлення працездатності вузла або сегмента корпоративної мережі (переустановлення системи, зміна конфігурації та ін.), грн;

V - втрати від зниження обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі, грн.

Втрати від зниження продуктивності співробітників атакованого вузла або сегмента корпоративної мережі являють собою втрати їхньої заробітної плати (оплата непродуктивної праці) за час простою внаслідок атаки:

$$П_п = \frac{\sum З_с}{F} * t_п = \frac{15000*1}{176} * 2 = 170{,}45 \text{ грн, де}$$

F - місячний фонд робочого часу (при 40-а годинному робочому тижні становить 176 ч).

Витрати на відновлення працездатності вузла або сегмента корпоративної мережі включають кілька складових:

$$П_в = П_{ви} + П_{пв} + П_{зч}, \text{ де}$$

$П_{ви}$ - витрати на повторне введення інформації, грн;

$П_{пв}$ - витрати на відновлення вузла або сегмента корпоративної мережі, грн.

$П_{зч}$ - вартість заміни устаткування або запасних частин, грн.

Витрати на повторне введення інформації $П_{ви}$ розраховуються виходячи з розміру заробітної плати співробітників атакованного вузла або сегмента корпоративної мережі $З_с$, які зайняті повторним введенням втраченої інформації, з урахуванням необхідного для цього часу $t_{ви}$:

$$П_{ви} = \frac{\sum З_с}{F} * t_{ви} = \frac{15000*1}{176} * 5 = 4261{,}36 \text{ грн}$$

Витрати на відновлення вузла або сегмента корпоративної мережі $П_{пв}$ визначаються часом відновлення після атаки $t_в$ і розміром середньогодинної заробітної плати обслуговуючого персоналу (адміністраторів):

$$П_{пв} = \frac{\sum З_о}{F} * t_в = \frac{0}{176} * 1 = 0 \text{ грн}$$

Втрати від зниження очікуваного обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі визначаються виходячи із середньогодинного обсягу продажів і сумарного часу простою атакованого вузла або сегмента корпоративної мережі:

$$V = \frac{О}{F_г} * (t_п + t_в + t_{ви}) = \frac{500000}{2080} * (2 + 1 + 5) = 1923{,}08 \text{ грн}$$

$F_г$ - річний фонд часу роботи організації (52 робочих тижні, 5-ти денний робочий тиждень, 8-ми годинний робочий день становить близько 2080 ч)

Вартість заміни устаткування або запасних частин може становити близько 3000 грн.

Вартість відновлення працездатності вузла або сегмента корпоративної мережі (переустановлення системи, зміна конфігурації та ін.):

$$П_в = 4261{,}36 + 0 + 3000 = 7261{,}36 \text{ грн}$$

$$U = 170{,}45 + 7261{,}36 + 1923{,}08 = 9254{,}89 \text{ грн}$$

Таким чином, загавоольний збиток від атаки на вузол або сегмент корпоративної мережі організації складе:

$$B = \sum_i \sum_n U = \sum_2 \sum_5 9254{,}89 = 92548{,}9 \text{ грн}$$

3.3.2 Загальний ефект від впровадження системи інформаційної безпеки

Загальний ефект від впровадження системи інформаційної безпеки визначається з урахуванням ризиків порушення інформаційної безпеки і становить:

$E = B * R - C$, де

В - загальний збиток від атаки на вузол або сегмент корпоративної мережі, тис. гри;

R - очікувана ймовірність атаки на вузол або сегмент корпоративної мережі, частки одиниці.

С - щорічні витрати на експлуатацію системи інформаційної безпеки, тис. гри.

$$E = 92548{,}9 * 0.5 - 0 = 46274{,}45 \text{ грн}$$

3.4 Визначення та аналіз показників економічної ефективності системи інформаційної безпеки

Оцінка економічної ефективності системи захисту інформації, розглянутої у спеціальній частині дипломного проекту, здійснюється на основі визначення та аналізу наступних показників:

а) сукупна вартість володіння (ТСО);

б) коефіцієнт повернення інвестицій (ROI). У сфері інформаційної безпеки йому відповідає показник ROSI (Return on Investment for Security);

в) термін окупності капітальних інвестицій $T_o$.

Показник сукупної вартості володіння (ТСО) використовується, якщо величину відверненого збитку від атаки на вузол або сегмент корпоративної мережі важко або неможливо визначити у вартісній формі.

Коефіцієнт повернення інвестицій ROSI показує, скільки гривень додаткового прибутку приносить одна гривня капітальних інвестицій на впровадження системи інформаційної безпеки. Щодо до інформаційної безпеки говорять не про прибуток, а про запобігання можливих втрат від атаки на сегмент або вузол корпоративної мережі, а отже

$$ROSI = \frac{E}{K}, \text{ частка одиниці, де}$$

Е - загальний ефект від впровадження системи інформаційної безпеки, тис. гри;

К - капітальні інвестиції за варіантами, що забезпечили цей ефект, тис. гри.

$$ROSI = \frac{46274,45}{63138,91} = 0,733, \text{ частки одиниці}$$

Проект визнається економічно доцільним, якщо розрахункове значення коефіцієнта повернення інвестицій перевищує величину річної депозитної ставки з урахуванням інфляції:

$ROSI > \frac{N_{\text{деп}} - N_{\text{інф}}}{100}$, де

$N_{\text{деп}}$- річна депозитна ставка або прибутковість альтернативного варіанту вкладення коштів, 6%;

$N_{\text{інф}}$- річний рівень інфляції, 5,5%.

$$0,733 > \frac{6 - 5,5}{100} = 0,733 > 0,005$$

Виходячи з розрахунків проведення дослідження є економічно доцільним.

Термін окупності капітальних інвестицій $T_o$ показує, за скільки років капітальні інвестиції окупляться за рахунок загального ефекту від впровадження системи інформаційної безпеки:

$$T_o = \frac{K}{E} = \frac{1}{ROSI} = \frac{1}{0{,}733} = 1{,}36 \text{ рік}$$

Висновки

У третій частині були проведені розрахунки з економічної доцільності дослідження та подальшому впровадженню Python бібліотек у процесі використання цифрової стегоранографії. Капітальні витрати складають 12636,48 грн. Щорічні експлуатаційні витрати - 0 грн. ROSI становить 0,733, а величина ефекту 46274,45 грн. Цей коефіцієнт перевищує величину річної депозитної ставки з урахуванням інфляції, і можна вважати дослідження та подальшому провадженню Python бібліотек у процесі використання цифрової стеганографії економічно доцільним. Термін окупності капітальних інвестицій становить 1,36 року.

# ВИСНОВКИ

У першому розділі кваліфікаційної роботи розглянуто місце використання цифрової стеганографії. Розглянуто основні методи стеганографічного захисту інформації та їх стегоанал.

У спец розділі проведено аналіз існуючих програмних засобів та відкритих бібліотек Python, що реалізують стеганографічні методи захисту інфрмації та методи стегоаналізу. Досліджено сумісність розглянутих бібліотек та програмного забезпечення.

В економічному розділі проведено розрахунок економічної доцільності дослідження Python бібліотек. Капітальні витрати складають 12636,48 грн. Експлуатаційні витрати - 0 грн/рік. Розрахований ROSI-коефіцієнт дорівнює 0,733. Термін окупності капітальних інвестицій становить 1,36 року.

ПЕРЕЛІК ПОСИЛАНЬ

1. Методичні вказівки до виконання економічної частини дипломного проекту зі спеціальності 125 Кібербезпека/Упорядн. Д. П. Пілова. - Дніпро: Національний технічний університет «Дніпровська політехніка», 2019.
2. Кваліфікаційна робота магістра. Методичні рекомендації до виконання для студентів спеціальності 125 «Кібербезпека» (освітньо-професійна програма «Кібербезпека») / Упоряд.: О.Ю.Гусєв, В.І.Корнієнко, В.І.Магро, Д.С. Тимофєєв; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Д.: НТУ «ДП», 2022. – 34 с.
3. Аналіз технік стеганографії текстових контенйерів. URL: https://www.researchgate.net/publication/306301164_Analysis_of_Different_Text_Steganography_Techniques_A_Survey
4. Стеганографія відео контейнеру. Останні дослідження та проблеми. URL: https://link.springer.com/article/10.1007/s11042-023-14844-w#Sec18
5. Порівняльне вивчення техніки цифрової аудіостеганографії. URL: https://asmp-eurasipjournals.springeropen.com/articles/10.1186/1687-4722-2012-25
6. XBOX 360 і стеганографія: як злочинці й терористи можуть «загаснути». URL: https://commons.erau.edu/cgi/viewcontent.cgi?article=1188&context=adfsl
7. Огляд стеганографії зображень. URL: http://martinolivier.com/open/stegoverview.pdf
8. Використання OpenStego. URL: https://www.openstego.com/
9. Техніка стеганографії за допомогою S-Tools. URL: https://medium.com/purple-team/steganography-technique-with-s-tools-6c82a52b0ed0
10. Stego-LSB. URL: https://github.com/ragibson/Steganography
11. Exstego. URL: https://github.com/v-gabriel/exstego
12. StegExpose. URL: https://github.com/b3dk7/StegExpose
13. LSB-Steganography. URL: https://github.com/RobinDavid/LSB-Steganography

Додаток А. Відомості матеріалів кваліфікаційної роботи

| | **Формат** | **Найменування** | **Кількість листів** | **Примітки** |
|---|---|---|---|---|
| | А4 | Реферат | 2 | |
| | А4 | Список умовних позначень | 1 | |
| | А4 | Зміст | 2 | |
| | А4 | Вступ | 1 | |
| | А4 | Стан питання. Постановка задачі | 31 | |
| | А4 | Спеціальна частина | 13 | |
| | А4 | Економічний розділ | 7 | |
| | А4 | Висновки | 1 | |
| | А4 | Перелік посилань | 1 | |
| 0 | А4 | Додаток А | 1 | |
| 1 | А4 | Додаток Б | 1 | |
| 2 | А4 | Додаток В | 1 | |
| 3 | А4 | Додаток Г | 1 | |
| 4 | А4 | Додаток Д | 1 | |

Додаток Г. Перелік матеріалів на оптичному носії

Дзеркаль_РА_125_22м_2_ПЗ.docx

Дзеркаль_РА_125м_22м_2_ПЗ.pdf

Дзеркаль_РА_125м_22м_2_ДМ.pptx

Додаток Д. Відгук керівника економічного розділу

Економічний розділ виконаний відповідно до вимог, які ставляться до кваліфікаційних робіт, та заслуговує на оцінку 95 б. («відмінно»).

Керівник економічного розділу _____ доц. Пілова Д. П.

(підпис) (ініціали, прізвище)

Додаток Е. Відгук керівника кваліфікаційної роботи

# ВІДГУК

**на кваліфікаційну роботу студента групи 125-22-2**

**Дзеркаля Романа Артуровича**

**на тему: «Дослідження особливостей Python бібліотек в процесі використання цифрової стеганографії»**

Пояснювальна записка складається зі вступу, трьох розділів і висновків, викладених на 56 сторінках.

Метою кваліфікаційної роботи є дослідження Python бібліотек у сфері цифрової стеганографії, порівняння та аналіз ефективності їх використання при розробці програмного забезпечення.

Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності магістра спеціальності 125 «Кібербезпека». Для досягнення поставленої мети в кваліфікаційній роботі вирішуються наступні задачі: аналіз існуючих методів стеганографії та стегоаналізу; розглянуто програмне забезпечення, що реалізує дані методи. Особливу увагу приділено Python бібліотекам, що знаходяться у відкритому доступі.

Практичне значення результатів кваліфікаційної роботи полягає у дослідженні особливостей реалізації обраних бібліотек та їх сумісності під час використання.

Оформлення пояснювальної записки до кваліфікаційної роботи виконано з незначними відхиленнями від стандартів.

За час дипломування Дзеркаль Р. А. проявив себе фахівцем, здатним самостійно вирішувати поставлені задачі та заслуговує присвоєння кваліфікації бакалавра за спеціальністю 125 Кібербезпека, освітньо-професійна програма «Кібербезпека».

Рівень запозичень у кваліфікаційній роботі не перевищує вимог "Положення про систему виявлення та запобігання плагіату".

Кваліфікаційна робота заслуговує оцінки 80 «добре».

**Керівник кваліфікаційної роботи** **Корнієнко В. І.**

**Керівник спец. розділу** **Тимофєєв Д. С.**