Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Пояснювальна записка
до магістерської кваліфікаційної роботи
«Магістр»
(освітньо-кваліфікаційний рівень)

на тему: **Підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра**

Виконав студент ІІ курсу, групи УБ – 20м
Спеціальність 125 – «Кібербезпека»
Освітня програма – «Управління інформаційною безпекою»
Софина Андрій Анатолійович
д.т.н., проф. каф. МБІС Яремчук Ю.Є.
«__»_____ 2021р.
Рецензент: к.т.н., доц. каф. ОТ Савицька Л.А.
«__»_____ 2021р.

Допущено до захисту
Голова секції УБ кафедри МБІС
д.т.н., проф. Яремчук Ю.Є.
“_____”_____2021 р.

Вінниця – 2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

АНОТАЦІЯ

В даній магістерській роботі проаналізовано та практично реалізовано програмний засіб для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

В роботі проведено теоретичний огляд обраної галузі, а саме особливості стеганографічних методів приховування інформації, їх класифікації, алгоритмів застосування в роботі та практичній реалізації.

В практичній частині роботи розроблено та обґрунтовано алгоритм роботи додатку, проектування користувацького інтерфейсу та програмна реалізація. Робота виконувалась на мові програмування C# в середовищі Visual Studio.

Ключові слова: стеганографія, стеганосистема, контейнер, схема розподілу Шаміра, метод найменш значущого біта, інформація, передача інформації.

SUMMARY

This master's thesis analyzes and practically implements a software tool to increase the stability and throughput of digital graphics containers by improving the steganographic method of NZB-infusion in combination with the distribution scheme of Shamir's secret.

The theoretical review of the chosen branch is carried out in the work, namely features of steganographic methods of hiding information, their classification, algorithms of application in work and practical realization.

In the practical part of the work the algorithm of the application operation, user interface design and software implementation are developed and substantiated. The work was performed in the C # programming language in Visual Studio.

Keywords: steganography, steganosystem, container, Shamir distribution scheme, least significant bit method, information, information transfer.

ЗМІСТ

ВСТУП

Актуальність. Стеганографія – це спосіб організації зв'язку, який приховує саме наявність зв'язку. На відміну від криптографії, де ворог точно може визначити, чи є повідомлення зашифрованим текстом, методи стеганографії дозволяють вбудовувати секретні повідомлення в невинні послання так, щоб неможливо було запідозрити існування вбудованого таємного послання.

Слово «стеганографія» у перекладі з грецької буквально означає "тайнопис" (steganos - секрет, таємниця; graphy - запис). До неї відноситься безліч секретних засобів зв'язку, таких як невидимі чорнила, мікрофотознімки, умовне розташування знаків, таємні канали і засоби зв'язку на плаваючих частотах і т.д.

Стеганографія займає свою нішу у забезпеченні безпеки: вона не замінює, а доповнює криптографію. Приховування повідомлення методами стеганографії значно знижує можливість виявлення самого факту передачі повідомлення. А якщо це повідомлення ще й зашифроване, воно має ще один, додатковий, рівень захисту.

В даний час у зв'язку з бурхливим розвитком обчислювальної техніки та нових каналів передачі інформації з'явилися нові стеганографічні методи, в основі яких лежать особливості представлення інформації в комп'ютерних файлах, обчислювальних мережах тощо. Це дає нам можливість говорити про становлення нового напряму – комп'ютерної стеганографії.

Таким чином, питання розроблення ефективних методів захисту цифрової інформації, зокрема методів комп’ютерної стеганографії та стеганоаналізу, актуальні та мають важливе значення для держави й суспільства.

Останнім часом велика кількість робіт присвячена технічному, криптографічному та стеганографічному захисту інформації, серед яких варто відзначити праці Хорошка В. О., Яремчука Ю. Є., Карпінця В. В., Шелеста М. Є., Конаховича В.Г. та ін. [1 – 5].

Мета дослідження. Метою роботи є розробка програмного засобу для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

Задачами дослідження є:

– здійснити аналіз обраної галузі, зокрема стеганографічних методів вбудовування даних у зображення;

– провести удосконалення та розробити алгоритм програмного засобу для підвищення стійкості та пропускної здатності цифрових графічних контейнерів;

– здійснити програмну реалізацію додатку для підвищення стійкості та пропускної здатності цифрових графічних контейнерів;

– обґрунтувати економічну доцільність розробки.

Об’єкт дослідження – процес захисту інформації, вкрапленої в графічний контейнер.

Предмет дослідження – методи та алгоритми комп’ютерної стеганографії і стеганоаналізу для зображень.

Новизна роботи – запропоновано алгоритм комп’ютерної стеганографії для цифрових контейнерів у вигляді зображення, що відрізняється підвищеною ефективністю, який дозволяє здійснювати операції вкраплення тексту на зображення.

1 АНАЛІЗ МЕТОДІВ СТЕГАНОГРАФІЧНОЇ ОБРОБКИ ДАНИХ ТА МЕТОДІВ ПЕРЕДАЧІ ІНФОРМАЦІЇ

1.1 Загальні поняття стеганографії та її практичне застосування

Стеганографія – це техніка приховування секретної інформації в каналі зв'язку таким чином, що приховується саме існування інформації. Це наука про «невидиму» комунікацію, яка не дозволяє ненавмисному одержувачу підозрювати, що дані існують. Стеганографія походить від грецького слова steganos, що означає «покритий», a graphy означає «пис», тобто закритий лист. Можна використовувати багато різних форматів носіїв файлів, таких як текст, зображення та відео, але цифрові зображення та аудіо є найпопулярнішими через їхню частоту в Інтернеті [6].

Стеганографія приховує різні типи даних у файлі обкладинки. Отриманий файл stego також містить приховану інформацію, хоча він практично ідентичний файлу обкладинки. Стеганографія використовує людське сприйняття; людські органи чуття не навчені шукати файли, які містять інформацію, приховану всередині них, хоча є програми, які можуть виконувати те, що називається Steganalysis (виявлення використання стеганографії). На рис.1[6] показана блок-схема безпечного стеганографічного системи.

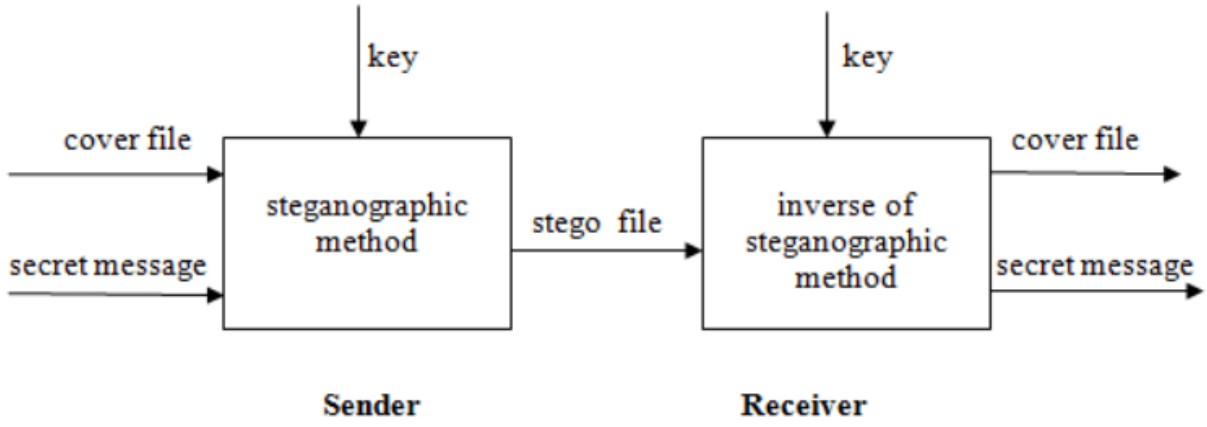


Рисунок 1.1 – Структура типової стеганографічної системи

Вхідними повідомленнями можуть бути зображення, тексти, відео тощо. Складовими стеганографічної системи є [7]:

– секретне повідомлення: секретне повідомлення або інформація, яку потрібно приховати.

– файл обкладинки/цифровий носій: дані або носій, який приховав секретне повідомлення.

– stego файл: змінена версія обкладинки, яка містить секретне повідомлення.

– ключ: додаткові секретні дані, які необхідні для процесів вбудовування та вилучення та повинні бути відомі як відправнику, так і одержувачу

– стеганографічний метод: стеганографічна функція, яка приймає прикриття, секретне повідомлення та ключ як параметри та створює stego як вихід.

Обернений стеганографічному методу: стеганографічна функція, яка має параметри stego та key та видає секретне повідомлення як вихід. Це зворотний метод, який використовується в процесі вбудовування в тому сенсі, що результат процесу вилучення є ідентичним вхідним параметрам процесу вбудовування.

Процес вбудовування вбудовує секретне повідомлення у файл обкладинки. Результатом функції вбудовування є дещо змінена версія файлу обкладинки: файл stego. Після того як одержувач отримав stego-файл, він починає процес вилучення з файлом stego та ключем як параметрами. Якщо ключ, наданий одержувачем, збігається з ключем, який використовується відправником для вбудовування секретного повідомлення, і якщо стего-дані, які одержувач використовує як вхідні дані, є тими ж даними, які створив відправник, то функція вилучення створить оригінальне секретне повідомлення [8 – 9].

Види стеганографії. В основному використовуються три типи стеганографічних протоколів: чиста стеганографія; стеганографія секретного ключа; стеганографія відкритого ключа.

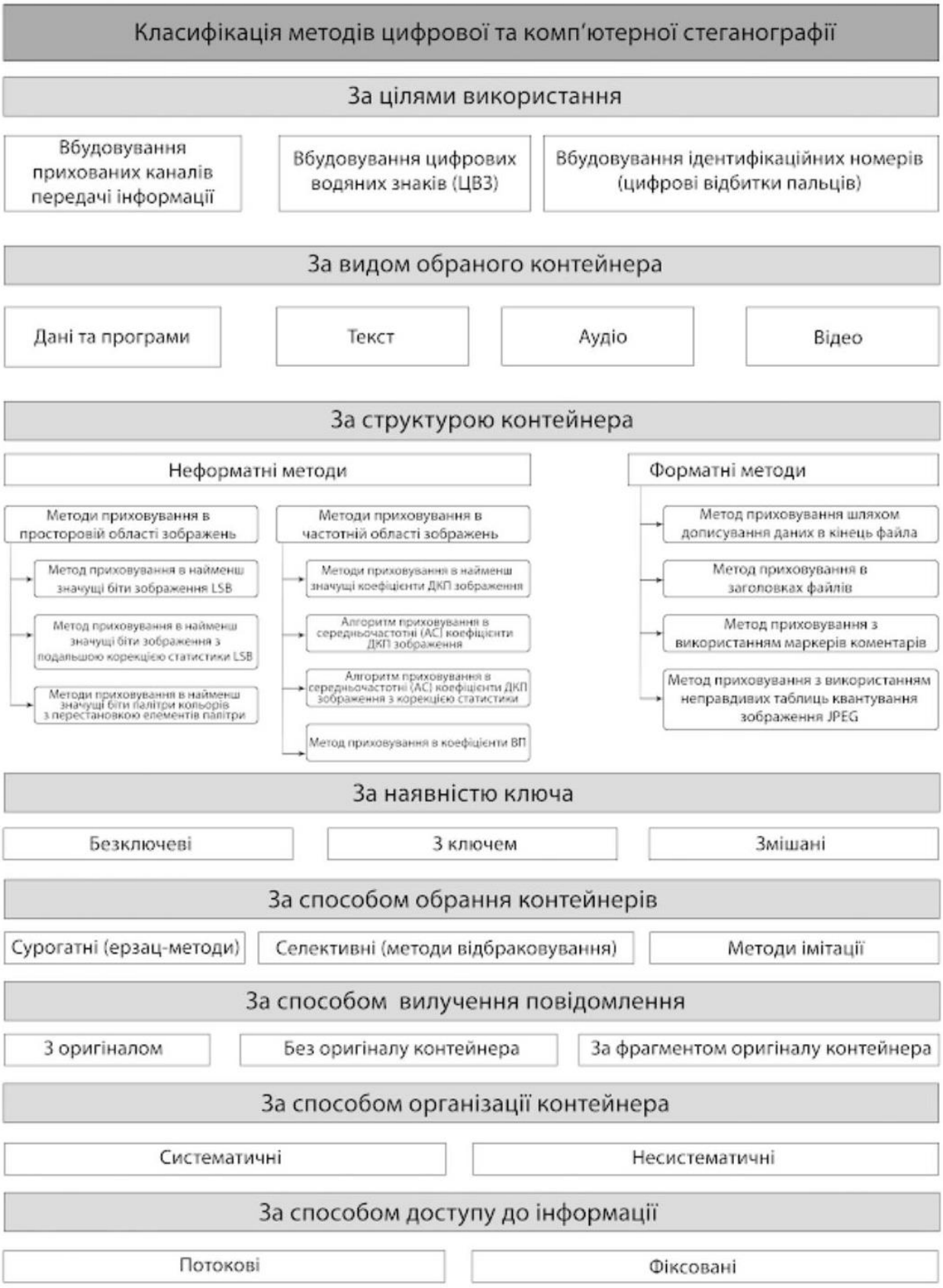


Рисунок 1.2 – Класифікація методів цифрової та комп'ютерної стеганографії (за [10])

Чиста стеганографія визначається як стеганографічна система, яка не вимагає обміну шифром, таким як стего-ключ. Цей метод не дуже безпечний, оскільки відправник і одержувач можуть покладатися лише на припущення, що жодна сторона не знає про секретне повідомлення [11].

Стеганографія секретного ключа визначається [12] як стеганографічна система, яка вимагає обміну секретним ключем (стего-ключем) перед спілкуванням. Секретний ключ стеганографії бере обкладинку і вбудовує секретне повідомлення всередину нього за допомогою секретного ключа (stego-key). Тільки сторони, які знають секретний ключ, можуть змінити процес і прочитати секретне повідомлення. На відміну від Pure Steganography, де є невидимий канал зв’язку, стеганографія із секретним ключем обмінюється стего-ключом, що робить його більш вразливим для перехоплення. Вигода від стеганографії секретного ключа навіть у разі її перехоплення; тільки сторони, які знають секретний ключ, можуть отримати секретне повідомлення.

Стеганографія відкритого ключа [13] визначається як стеганографічна система, яка використовує відкритий ключ і закритий ключ для захисту зв'язку між сторонами, які бажають спілкуватися таємно. Відправник буде використовувати відкритий ключ під час процесу кодування, і тільки закритий ключ, який має прямий математичний зв’язок з відкритим ключем, може розшифрувати секретне повідомлення. Стеганографія відкритого ключа забезпечує більш надійний спосіб реалізації стеганографічної системи, оскільки вона може використовувати набагато більш надійну та досліджену технологію криптографії з відкритим ключем. Він також має кілька рівнів безпеки, оскільки небажані сторони повинні спочатку запідозрити використання стеганографії, а потім їм доведеться знайти спосіб зламати алгоритм, який використовується системою відкритих ключів, перш ніж вони зможуть перехопити секретне повідомлення.

Характеристики стенографічної системи [14]. Ефективна стеганографічна схема повинна мати такі бажані характеристики:

– секретність: особа не повинна мати можливість витягти приховані дані

з хоста без знання належного секретного ключа, який використовується в процедурі вилучення.

– невідчутність: носій після вбудовування прихованих даних повинен бути непомітним від вихідного носія. Не слід сумніватися в існуванні прихованих даних усередині носія.

– висока ємність: максимальна довжина прихованого повідомлення, яке можна вставити, має бути якомога довшою.

– стійкість: секретні дані повинні бути в змозі вижити, коли хост-носій маніпулював, наприклад, за допомогою якоїсь схеми стиснення з втратами.

– точне вилучення: вилучення прихованих даних із носія має бути точним та надійним.

У стеганографії, у якості носія, можна використовувати чотири основні категорії форматів файлів: текст; зображення; аудіо/відео; протокол [15].

Стеганографія тексту: приховування інформації в тексті є найважливішим методом стеганографії. Метод полягає у тому, щоб приховати секретне повідомлення в кожній n-й літері кожного слова текстового повідомлення. Після розвитку Інтернету та різних типів цифрових форматів файлів його значення зменшилося. Текстова стенографія з використанням цифрових файлів використовується не дуже часто, оскільки текстові файли мають дуже невелику кількість зайвих даних

Стеганографія зображення [16]: зображення використовуються як популярні об’єкти для стеганографії. Повідомлення вбудовується в цифрове зображення за допомогою алгоритму вбудовування з використанням секретного ключа. Отримане стего-зображення надсилається одержувачу. З іншого боку, він обробляється алгоритмом вилучення за допомогою того ж ключа. Під час передачі стего-зображення неавтентифіковані особи можуть лише помітити передачу зображення, але не можуть здогадатися про існування прихованого повідомлення.

Аудіостеганографія: аудіо стенографія [17] – це маскування, яке використовує властивості людського вуха, щоб непомітно приховувати

інформацію. Чутний звук може бути нечутним за наявності іншого гучнішого чутного звуку. Ця властивість дозволяє вибрати канал, у якому потрібно приховати інформацію.

Стеганографія протоколу [18]: Термін стеганографія протоколу означає вбудовування інформації в мережеві протоколи, такі як TCP/IP. Ми приховуємо інформацію в заголовку пакета TCP/IP в деяких полях, які можуть бути необов'язковими або ніколи не використовуватися.

Отже, призначення не замінити криптографію, а доповнити її. Приховування повідомлення за допомогою методів стеганографії зменшує ймовірність виявлення повідомлення.

Однак, якщо це повідомлення також зашифроване, якщо воно виявлено, воно також може бути зламане. Існує нескінченна кількість додатків стеганографії.

Стеганографія стосується не тільки цифрових зображень, а й інших медіа (файли, такі як голос, інший текст і двійкові файли; інші медіа, такі як канали зв'язку, список можна продовжувати до нескінченності).

Стеганографія не є хорошим рішенням для секретності. Якщо повідомлення зашифровано за допомогою підстановки (заміна одного алфавіту іншим), перестановки повідомлення (перетасування тексту) і повторного застосування заміни, тоді зашифрований зашифрований текст є більш безпечним, ніж використання лише заміни або лише перестановки.

Зараз, якщо зашифрований текст вбудований у зображення, відео, голос тощо, це ще більш безпечно. Якщо зашифроване повідомлення перехоплено, перехоплювач знає, що текст є зашифрованим повідомленням. За допомогою стеганографії перехоплювач може не знати, що об'єкт містить повідомлення [19].

1.2 Аналіз контейнерів для стеганографічної обробки

Перш ніж, встановити класифікацію, важливо представити типи зображень, вибраних для нашої класифікації [20].

Обрано найбільш часто використовувані формати, які зараз використовуються та обмінюються. Необхідно розрізняти різні уявлення зображень. У файлі для зберігання та обміну даними зображення зазвичай стискається і зберігається у графічному форматі.

Основними матричними форматами є растрове зображення Windows (BMP), формат обміну графікою (GIF) [21], портативна мережева графіка (PNG) [22] і Joint Photographic Experts Group (JPEG) [23].

Кожен формат має свої особливості. Щоб вибрати той, який відповідає тому, що ми хочемо зробити з нашими зображеннями, важливо знати глибину кольорів. Виражений у бітах, він відповідає кількості значень кольору, які може прийняти кожен піксель зображення.

Формат BMP (або Bitmap) [24] – це універсальний нестиснений формат, розроблений Microsoft та IBM. Це дозволяє точно відтворювати кольори вихідного зображення; відповідником є велика вага згенерованого файлу.

Формат обміну графікою (GIF): також широко використовується в Інтернеті, цей власний формат використовує індексовану колірну схему. Таким чином, можна використовувати лише конкретні значення кольору, що дає можливість максимально оптимізувати вагу візуального. На відміну від цього, зовнішній вигляд візуальних зображень, що відображають багато кольорів, сильно погіршується. Цей формат також дозволяє створювати покадрову анімацію.

Формат Joint Photographic Experts Group (JPEG) [23]: широко використовується в Інтернеті, цей формат був розроблений групою експертів, яка публікує стандарти стиснення фотографій. Цей стиснутий формат значно змінює якість вихідних зображень, але забезпечує відносно точне відтворення кольору та відносно невелику вагу.

Формат Portable Network Graphics (PNG) [22, 25]: цей формат стандартизований Консорціумом World Wide Web Consortium (W3C) і призначений для обходу існуючого власного формату GIF. Таким чином, PNG має точно такі ж характеристики, як і GIF, а також дозволяє записувати від 1 до 48 біт і керувати прозорістю (альфа-канали). З іншого боку, неможливо створити анімацію.

1.3 Класифікація методів цифрової стеганографії

Класифікація стеганографічних методів представлена на рисунку 1.2. Як можна поачити, стеганографічні методи можна вивчати у п'яти групах: методи злиття, техніка статистичної модифікації, техніка перестановки, заміни та адитивного маркування [26].

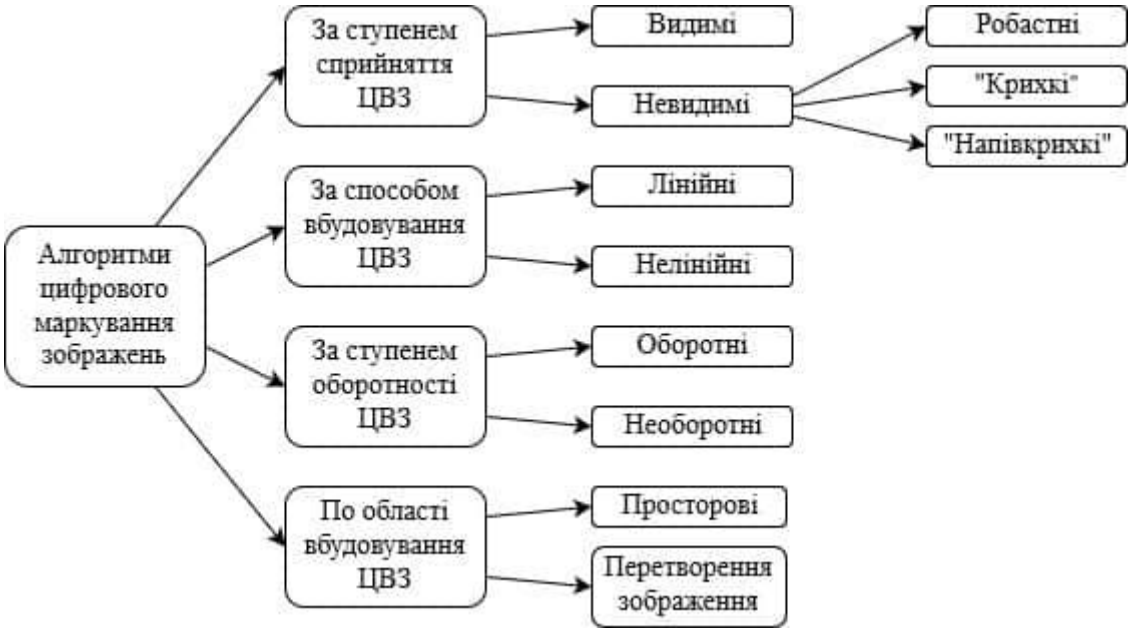


Рисунок 1.3 – Методи цифрової стеганографії (за [27])

Fusion: Цей метод, який можна вважати наївною стеганографією [28], полягає в додаванні даних, які потрібно приховати, до файлу. Для цього цей метод використовує невикористані або непрочитані слоти більшістю декодерів зображень, є дві операції: додавання даних у кінець файлу та додавання до заголовків файлу.

Додавання в кінці файлу стало можливим завдяки тому, що більшість декодерів зображень не зчитують файл зображення як єдине ціле. Для більшості доступних форматів зображень певний бітовий рядок встановлюється для позначення кінця зображення [29].

Додавання в кінці зображення просто додає приховані дані після цього рядка. Немає обмежень за розміром; однак файл зображення розміром 20 Мбайт не може залишитися непоміченим. Що стосується додавання до заголовка, то деякі формати, такі як растрове зображення, визначають поле для визначення зміщення, з якого почнеться зображення. Вказавши трохи довше зміщення, можна приховати дані, які будуть приховані між двома зміщеннями.

Статистичні методи: Статистичні методи модифікують декілька допоміжних статистичних даних (частоти літер, розподіл пікселів), щоб приховати повідомлення [30] та отримати його, перевіривши ці припущення. У цій техніці створена позначка безпосередньо вставляється в вихідне зображення. Процес виявлення та вилучення в цьому типі стеганографії (для сліпих методик) здійснюється за допомогою статистичних методів. Наприклад, можна виконати кореляційне вимірювання.

Адитивні схеми: під час вставки сигнал, що представляє позначку, додається до певних компонентів середовища. Для цього не потрібно ні адаптувати марку до середовища, щоб сигнал, який він представляє, не був занадто низьким (ризик невиявленості), ні проблемами надійності, ні занадто сильними (стирання початкового сигналу і, отже, занадто сильним). велика його деградація).

Розширення спектру [31] – це техніка, яка використовується в радіотелекомунікації, особливо у військових, для розсіювання сигналу в широкій смузі частот, щоб зробити його стриманим і стійким до перешкод. Рисунок 6: Класифікація стеганографічних технік за форматом зображення

147 ISSN: 2456-7132 Доступний онлайн на Journals.aijr.in Khaldi Amine, Int. Енн наук.; Vol. 8, випуск 1, стор. 143-149, 2020 р. Рисунок 7: Класифікація технік заміни. Техніка заміни: у методах заміни інформація, яку потрібно

приховати, не додається, а замінюється компонентами зображення (пікселем, коефіцієнтом перетворення), вибраними за допомогою секретний ключ. Техніку заміщення можна розділити на дві підкатегорії, прямі та непрямі. При прямій заміні значення покриття змінюються без будь-якого процесу перетворення. При непрямій заміні перед процесом приховування проводиться трансформація покриття [32].

Сегментація складності на бітовій площині (BPCS): зорова система людини має таку особливу властивість, що занадто складний візуальний шаблон не може сприйматися як «інформативний про форму» [33] Наприклад, на дуже рівному березі пляжу кожен квадрат- зона підніжжя виглядає так само - це просто піщана ділянка, форми не спостерігається. Однак, якщо придивитися уважно, дві однакові на вигляд області абсолютно різні за формою частинок піску. BPCS-Steganography використовує цю властивість. Він замінює складні області на бітових площинах зображення судна іншими складними шаблонами даних (тобто частинами секретних файлів). Ця операція заміни називається «вбудовуванням». Ніхто не може побачити різницю між двома зображеннями судин до та після операції вбудовування.

Модифікація коефіцієнтів DCT: цей підхід полягає у виділенні певної кількості квадратів розміром 8 × 8 пікселів із зображення, обчисленні DCT-перетворення цих блоків і відмічанні біта на середніх частотах, модифікації низьких частот зображення не змінив би це занадто сильно. Низькі частоти, що відповідають найбільшим однорідним ділянкам на зображенні, наприклад, рівномірний чорний у темних областях, і високі частоти, які видаляються за допомогою стиснення JPEG, відповідають найменшим однорідним ділянкам зображення [34].

Найменший біт (LSB): LSB збирає все, що стосується приховування даних, змінюючи нижчий біт елемента [35]. Зміна значення пікселя або зміна значення коефіцієнта DCT у випадку стандарту JPEG. Всі вони засновані на нечутливості зорової системи людини до невеликих змін кольорів. Існує два методи LSB:

Заміна LSB: Ця техніка полягає у заміні найменших значущих бітів пікселів на біти повідомлення, які потрібно вставити. Щоб вставити повідомлення, останній найменший біт кожного пікселя замінюється на біт повідомлення, яке потрібно приховати. Напрямок шляху пікселів зазвичай вибирається псевдовипадковим шляхом. Для цього передавач і приймач повинні спочатку обмінятися ключем k, який використовується як початковий генератор псевдовипадкових чисел.

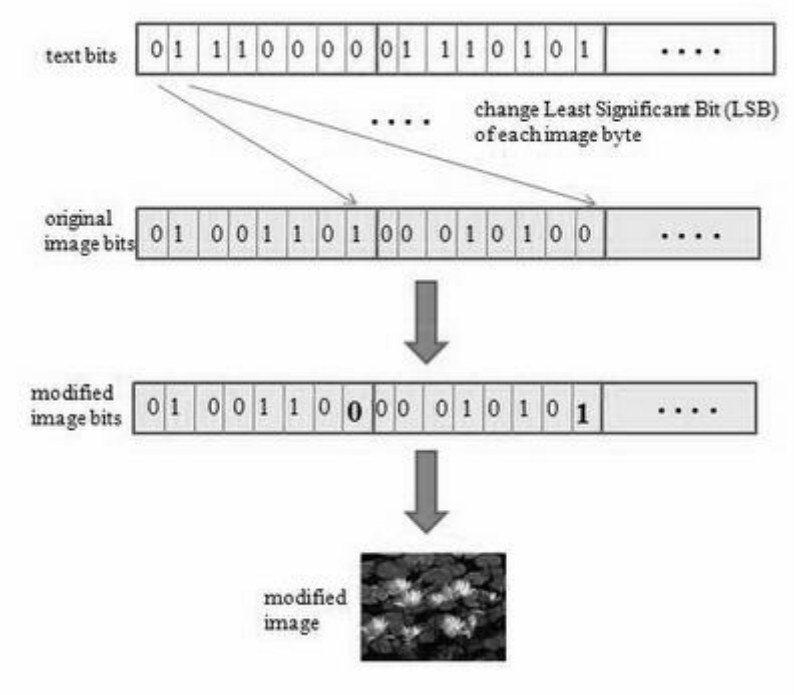


Рисунок 1.4 – Представлення методу LSB (за [36 – 37])

Стеганографія узгодження LSB: стеганографія узгодження з найменшим значущим бітом (LSB), яка також називається вбудовуванням ±1, є дещо більш складною версією вбудовування LSB. Метод стеганографії кореспонденції LSB не змінює статистичний розподіл першого порядку підтримки хоста. Отже, всі статистичні атаки першого порядку неефективні [38]. Стеганографічну техніку не завжди можна застосувати до всіх форматів зображень. Наприклад, неможливо змінити коефіцієнти DCT зображення BMP, оскільки цей формат не стискається, також неможливо змінити LUT в зображенні BMP, оскільки колориметричне представлення в зображенні BMP не використовує таблицю

пошуку. . Для зображень JPEG, PNG і GIF пряма заміна не може бути виконана, оскільки обидва типи потребують обробки для доступу до значень пікселів (DCT для JPEG і LUT для GIF).

1.4 Аналіз атак на стеганографічні системи

Розглянемо класифікацію атак порушника, який намагається визначити факт прихованої передачі повідомлення та при встановленні цього факту намагається переглядати їх.

Рисунок 1.5 – Атаки на стегосистему (за [39 – 40])

Атака лише зі стегограмою. Порушнику відома одна або кілька стегограм і він намагається визначити, чи не містять вони прихованих повідомлень, і якщо так, то намагається читати їх.

Порушнику дуже важко зламати стегосистему у цій атаці. Це пояснюється тим, що при невідомості ні вихідного контейнера, ні будь-якої частини приховуваного повідомлення можна отримати дуже велику кількість помилкових розшифровок, серед яких не можна віддати перевагу. Девід Кан у своїй знаменитій книзі описував, що якщо цензор при перегляді поштових відправлень у роки Другої світової війни не міг відразу знайти слідів повідомлень, що приховуються, то швидше за все це завдання не має однозначного рішення [41].

Атака із відомим контейнером. Порушнику доступна одна або безліч пар контейнерів та відповідних їм стегограм. Зауважимо, що у цій атаці порушник знає вихідний вид контейнера, що дає йому істотні переваги проти першої атакою. Наприклад, як відомий порушник контейнера може служити студійний запис музичного твору, яке передається по радіомовному каналу з вбудованою інформацією. Або як контейнер використовується зображення будь-якої відомої картини, що демонструється в Ермітажі, високоякісна цифрова копія якої вільно продається на CD-дисках [42].

Атака із вибраним контейнером. Порушник здатний нав'язати для використання в стегосистемі конкретний контейнер, який має якісь переваги для проведення стегоаналізу в порівнянні з усією безліччю контейнерів. Удосконалена версія цієї атаки: атака з контейнерами, що адаптивно вибираються. Порушник нав'язує контейнер, аналізує отримане стего для формування оцінок ймовірності факту прихованої передачі або оцінок приховуваного повідомлення або оцінок стегоключа, що використовується. На основі отриманих оцінок порушник формує черговий контейнер, з урахуванням чергового стего уточнює оцінки і так далі до однозначного встановлення факту наявності прихованого зв'язку або його відсутності, а при виявленні каналу прихованого зв'язку до обчислення стегоключа, що використовується, і читання

прихованого листування.

Наприклад, така атака може мати місце при несанкціонованому використанні відправником повідомлень чужого каналу передачі інформації, що приховуються, коли законний власник інформаційних ресурсів проводить розслідування з метою позбутися непроханих користувачів. Зокрема, в сучасних телекомунікаційних системах відомі спроби безкоштовно скористатися послугами дорогого супутникового та наземного мобільного зв'язку [43].

Атака із відомим повідомленням. Порушнику відомий зміст одного або декількох повідомлень, що приховуються, і він намагається встановити факт їх передачі і/або стогоключ. Наприклад, така атака виконується тюремником Віллі у класичному завданні про ув'язнених [44]. Віллі, знаючи вид повідомлення про втечу, аналізує листування між ув'язненими, щоб виявити момент втечі, що готується. Зрозуміло, що знайти сліди конкретного повідомлення у певному безлічі переданих стего значно простіше, ніж виявити у цьому безлічі факт прихованої передачі апріорі невідомого повідомлення.

Якщо порушнику відомі деякі приховувані повідомлення та відповідні їм стегограми, то його завданням є визначення ключа стегосистеми для виявлення та читання інших повідомлень, що приховуються, або при неможливості (високій складності) визначення ключа завданням порушника є побудова методів безключового читання або визначення факту передачі приховуваної інформації.

Атака із вибраним повідомленням [45]. Порушник здатний нав'язати передачі по стегосистеме конкретне повідомлення і намагається встановити факт його прихованої передачі та використовуваний секретний ключ. Також можлива атака з адаптивно вибираються повідомленнями, в якій порушник послідовно підкидає приховує інформацію підбираються повідомлення і ітеративно зменшує свою невизначеність про використання стегосистеми та її параметри.

Наприклад, така атака може виконуватися, коли виникає підозра, що з

якогось автоматизованого робочого місця (АРМ) локальної мережі установи відбувається витік конфіденційної інформації, яка потім потай передається за межі цієї мережі. Для виявлення каналу витоку адміністратор безпеки формує повідомлення, які б зацікавити несумлінного користувача і вводить їх у інформаційні масиви мережі. Потім адміністратор намагається виявити сліди цих повідомлень в інформаційних потоках, які передаються з АРМ користувачів через сервер у зовнішні мережі. Для однозначного встановлення факту наявності або відсутності каналу прихованого зв'язку адміністратор вибирає такі повідомлення, які найлегше виявити при їх передачі по стегоканалу.

Крім того, можливі різні поєднання перерахованих атак, у яких порушник здатний знати або вибирати використовувані контейнери та скритно передані повідомлення. Ступінь ефективності атак на стегосистему зростає в міру збільшення знань порушника про використовувані контейнери, повідомлень, що приховуються, обсягу перехоплених стегограм і його можливостей з нав'язування обраних контейнерів і повідомлень [46].

Введемо моделі порушника, який намагається протидіяти прихованню інформації. Наслідуючи К. Шеннону, назвемо першу з цих моделей теоретико-інформаційної [47]. Нехай, як це прийнято для систем захисту інформації, для стегосистем виконується принцип Кергоффа: порушник знає повний опис стегосистеми, йому відомі імовірнісні характеристики повідомлень, що приховуються, контейнерів, ключів, формуються стегограм. Порушник має необмежені обчислювальні ресурси, що запам'ятовують пристроями довільно великої ємності, має в своєму розпорядженні нескінченно великий час для стегоаналізу і йому відомо довільно безліч перехоплених стегограм [48]. Єдине, що невідомо порушнику - ключ стегосистеми, що використовується. Якщо в даній моделі порушник не в змозі встановити, міститься чи ні приховуване повідомлення в стезі, що спостерігається, то назвемо таку стегосистему теоретико-інформаційно стійкою до атак пасивного порушника або досконалої.

Стійкість різних стегосистем може бути розділена на стійкість до

виявлення факту передачі (існування) інформації, що ховається, стійкість до вилучення приховуваної інформації, стійкість до нав'язування хибних повідомлень по каналу прихованого зв'язку, стійкість до відновлення секретного ключа стегосистеми.

Очевидно, що якщо стегосистема є стійкою до виявлення факту передачі (існування) інформації, що приховується, то логічно припустити, що вона при цьому є стійкою і до читання інформації, що приховується. Зворотне у випадку неправильне. Стегосистема може бути стійкою до читання інформації, що приховується, але факт передачі деякої інформації під прикриттям контейнера може виявлятися порушником. Перефразовуючи відоме висловлювання Ш.Гольдвассера про несиметричні системи шифрування [49], можна сказати, що якщо накрити верблюда ковдрою, то можна приховати число горбів у верблюда (назвемо це прихованим повідомленням), але важко приховати, що під ковдрою-контейнером щось приховано .

Стійкість стегосистеми до нав'язування хибних повідомлень по каналу прихованого зв'язку характеризує її здатність виявляти і відкидати сформовані порушником повідомлення, що вводяться в канал передачі повідомлень, що приховуються, з метою видачі їх за істинні, що виходять від законного відправника. Наприклад, якщо в класичному завданні Сіммонса про ув'язнених тюремник Віллі виявиться здатним сфабрикувати хибне повідомлення про відміну втечі і одержувач Боб повірить, що її автором є законний відправник Аліса, то це означає істотну слабкість стегосистеми, що використовується.

Якщо в системі ЦВЗ зловмисник здатний ввести в контейнер, завірений законним відправником, свій водяний знак і детектор виявлятиме водяний знак зловмисника і не виявлятиме ЦВЗ справжнього відправника, то це означає дискредитацію (злом) системи ЦВЗ [50]. Стійкість до відновлення секретного ключа стегосистеми характеризує її здатність протистояти спробам порушника обчислити секретну ключову інформацію цієї стегосистеми. Якщо порушник здатний визначити ключ симетричної стегосистеми, він може однозначно виявляти факти передачі повідомлень, що приховуються, і читати їх або

нав'язувати хибні повідомлення без будь-яких обмежень. Таку подію можна назвати повною компрометацією стегосистеми. Очевидно, що атаки порушника на ключ стегосистеми можуть бути побудовані аналогічно атакам на ключ систем шифрування інформації та систем аутентифікації повідомлень.

Якщо порушник здатний обчислити ключ вбудовування водяного знака будь-якого автора (власника) інформаційних ресурсів, він може поставити цей водяний знак будь-який контейнер. Тим самим порушник дискредитує або водяний знак даного автора (власника), або повністю всю систему ЦВЗ. В обох випадках ставиться під сумнів законність прав одного або всіх власників інформаційних ресурсів на те, що справді належить.

1.5 Висновки та постановка задач

Отже, в даному розділі було проведено теоретичний огляд галузі, в якій проводиться розробка. У даній роботі представлено основні поняття стеганографічних систем передачі інформації, також проведено аналіз класифікації стеганографічних методів та можливих атак на них.

В результаті проведеного аналізу теоретичного матеріалу та виходячи з мети і актуальності теми, були поставлені наступні задачі подальшої роботи:

– здійснити аналіз обраної галузі, зокрема стеганографічних методів вбудовування даних у зображення;

– провести удосконалення та розробити алгоритм програмного засобу для підвищення стійкості та пропускної здатності цифрових графічних контейнерів;

– здійснити програмну реалізацію додатку для підвищення стійкості та пропускної здатності цифрових графічних контейнерів;

– обґрунтувати економічну доцільність розробки.

В результаті виконання поставлених завдань, планується досягти основної мети роботи, а саме розробки та реалізації програмного засобу для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

2 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ НА ОСНОВІ МЕТОДУ НЗБ ТА СХЕМИ РОЗПОІДЛУ СЕКРЕТУ ШАМІРА

2.1 Удосконалення методу найменш значущого біта

Цифрові дані обчислюються у двійковому форматі, і, як і числові записи, права цифра вважається найнижчою цифрою, а крайня ліва — найвищою. Завдяки позиційній нотації, молодший біт також відомий як крайній правий біт. Це протилежність старшого біта, який несе найвище значення в багаторазрядному двійковому чисі, а також число, яке знаходиться найдальше праворуч. У багаторозрядному двійковому числі значущість біта зменшується, коли він наближається до молодшого біта. Оскільки він двійковий, старший біт може бути 1 або 0 [51 – 52].

Коли передача двійкових даних здійснюється за допомогою методу першого младшого значущого біта, першим передається найменший значущий біт, за яким слідують інші біти зростаючого значення. Найменший біт часто використовується в хеш-функціях, контрольних сумах і генераторах псевдовипадкових чисел.

Суть методу заміна найменш значущого біта (Least Significant Bits – LSB) полягає в приховуванні інформації шляхом зміни останніх бітів зображення, що кодують колір на біти приховуваного повідомлення. Різниця між порожнім та заповненим контейнерами має бути не відчутною для органів сприйняття людини. Він є найпоширенішим у електронній стеганографії.

Ґрунтується на обмежених здібностях органів чуття, внаслідок чого людям дуже важко розрізняти незначні варіації звуку чи кольору.

Розглянемо цей метод з прикладу 24 бітного растрового RGB зображення. Кожна точка кодується 3 байтами, кожен байт визначає інтенсивність червоного (Red), зеленого (Green) та синього (Blue) кольору [53 – 54].

Сукупність інтенсивностей кольору в кожному з трьох каналів визначає відтінок пікселя. Змінюючи найменш значущий біт, ми змінюємо значення байт

на одиницю. Такі градації, крім того, що непомітні для людини, можуть взагалі не відобразитись при використанні низькоякісних пристроїв виведення.

Наведений нижче приклад показує, як повідомлення може бути приховане по-перше восьми байтах, що відносяться до трьох пікселів у 24-бітного зображення.

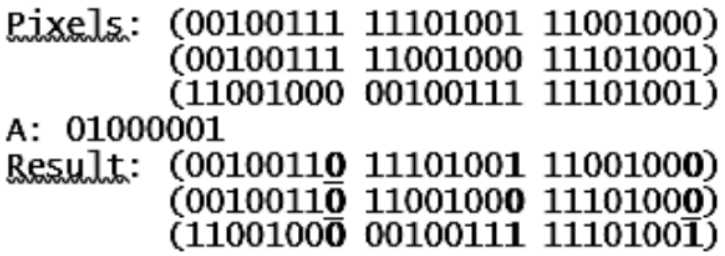


Рисунок 2.1 – Приклад роботи методу найменш значущого біта для 24 бітного растрового RGB зображення (за [55])

Стегоаналіз методу LSB. Порушення статистичних закономірностей природних контейнерів є одним із найбільш перспективних підходів виявлення факту існування прихованого каналу передачі є підхід, що представляє введення в файл інформації, що приховується. При даному підході аналізуються статистичні характеристики досліджуваної послідовності та встановлюється, чи схожі вони на характеристики природних контейнерів (якщо так, то прихованої передачі немає), або вони схожі на характеристики стего (якщо так, то виявлено факт існування прихованого каналу передачі). Цей клас стеганоатак є імовірнісним, тобто вони не дають однозначної відповіді, а формують оцінки типу «дана досліджувана послідовність з ймовірністю 90% містить приховуване повідомлення» [56].

У методі використовується аналіз гістограми, отриманої за елементами зображення та оцінка розподілу пар значень цієї гістограми. Для BMP файлів пари значень формуються значеннями пікселів зображення, для JPEG квантуються коефіцієнтами дискретного косинусного перетворення, які відрізняються за молодшим бітом. Молодші біти зображень не є випадковими. Частоти двох сусідніх елементів контейнера мають бути досить далеко від значення частоти середнього арифметичного цих елементів.

В «порожньому» зображенні ситуація, коли частоти елементів зі значеннями 2N та 2N + 1 близькі за значенням, трапляється досить рідко. При вбудовуванні інформації дані частоти зближуються або стають рівними.

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

На рисунку 2.2 зображено декомпозицію пікселя на складові в просторі RGB (три байти) та позначено найменш значущі біти (LSB), що будуть замінені на біти прихованого повідомлення.

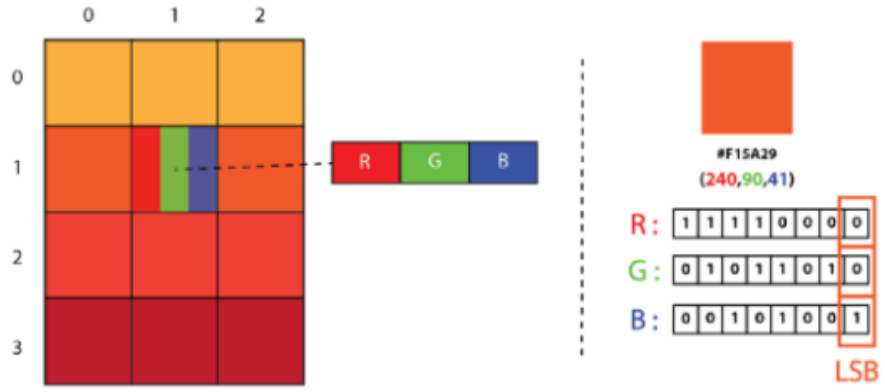


Рисунок 2.2 – Найменш значущі біти в пікселі зображення (за [57])

Основною перевагою методу найменш значущого біта є те, що він дозволяє вбудовувати достатньо великі об’єми інформації в невеликі файли (пропускна спроможність прихованого каналу інформації складає від 12,5 до 30%) [3].

Недоліком НЗБ є низька стеганографічна стійкість – найменші спотворення контейнера можуть призвести до втрати повідомлення або його частини. Ще одним недоліком методу є те, що при перегляді зображення з великим масштабом чітко видно області з вбудованим повідомленням. Та все ж даний метод є достатньо популярним зважаючи на його простоту в реалізації.

Для удосконалення методу найменш значущого біта розглянемо приклад: нехай, є 24-х бітове зображення в градаціях сірого. Кожен піксель такого зображення кодується 3 байтами, в яких розташовані значення каналів RGB. Здійснюючи заміну найменш значущого біта, значення байта змінюється на одиницю [58].

Такі зміни можна вважати удосконаленням, оскільки змінені градації зображення фактично не помітні для людського зору, проте, досить вдалі для розпізнавання стеганографічними системами.

Така часткова модифікація даного стеганографічного методу надає можливість використовувати для вбудовування повідомлення два або більше молодших бітів на байт. Це збільшує обсяг прихованої інформації в об'єкті-контейнері, проте приховуваність помітно сильно знижується, що полегшує процес розпізнавання стеганографії. Інші варіації цього методу включають в себе нівелювання статистичних змін в зображенні.

2.2 Алгоритм роботи схеми розподілу секрету Шаміра

Завдання поділу секрету має на увазі поділ секретної інформації між учасниками так, що тільки заздалегідь задані безлічі учасників зможуть її відновити, отже, ймовірність компрометації цієї інформації знижується.

Наочним прикладом може бути кімната, в якій зберігається щось цінне для певної групи осіб. Якщо закрити кімнату на кілька різних між собою замків, кількість яких дорівнює кількості учасників, і роздати кожному члену цієї групи за одним ключем, то відкрити двері зможуть тільки всі учасники, зібравшись разом.

Ідея, на якій заснована схема Шаміра, полягає в тому, що для інтерполяції многочлена ступеня k-1 потрібно k точок. Якщо відомо меншу кількість точок, то інтерполяція буде неможливою [59 – 60].

Позначимо: p - велике просте число (більше будь-якого секрету M, який передбачається розділяти в цій схемі). Тоді $M \in Z_p: n,$ n – число часток секрету; k - мінімальний розмір дозволеної групи.

Роботу алгоритму можна розділити на 3 етапи.

Підготовчий етап.

Дилер вибирає випадковим чином коефіцієнти секретний многочлен:

$$S_1, S_2, S_3, \dots S_{k-1} \in Z_p$$

і складає

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

де M – розділяючий секрет, а інші коефіцієнти - довільні елементи поля (коефіцієнти многочлена дилер зберігає в таємниці). очевидно, що S(0) = M. Далі дилер обирає n різних несекретних ненульових елементів $r_1, r_2, r_3, \dots, r_n$ із Z_p, відповідність одному учаснику схеми.

Розподіл секрету [61].

Дилер обчислює значення наступного многочлена:

$$c_1 = S(r_1), c_2 = S(r_2), \dots, c_n = S(r_n)$$

Частка кожного користувача Ai - це пара чисел $(r_i, c_i), i = 1,2,\dots,n$. Частки роздають учасникам схеми.

Відновлення секрету [62].

Щоб відновити секрет, треба скористатися інтерполяційною формулою Лагранжа: якщо потрібно побудувати многочлен S(x) ступеня k-1, який при $x_1, x_2, \dots, x_k$ приймає відповідно значення $y_1, y_2, \dots, y_k$, то цим многочленом буде:

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

k як в схемі розподілу секрету многочлен належить обрати так, щоб S (0) = M, то з формули Лагранжа слідує:

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

З описаного вище, стає ясно, що для більших значень порога, обчислення стає повільнішим.

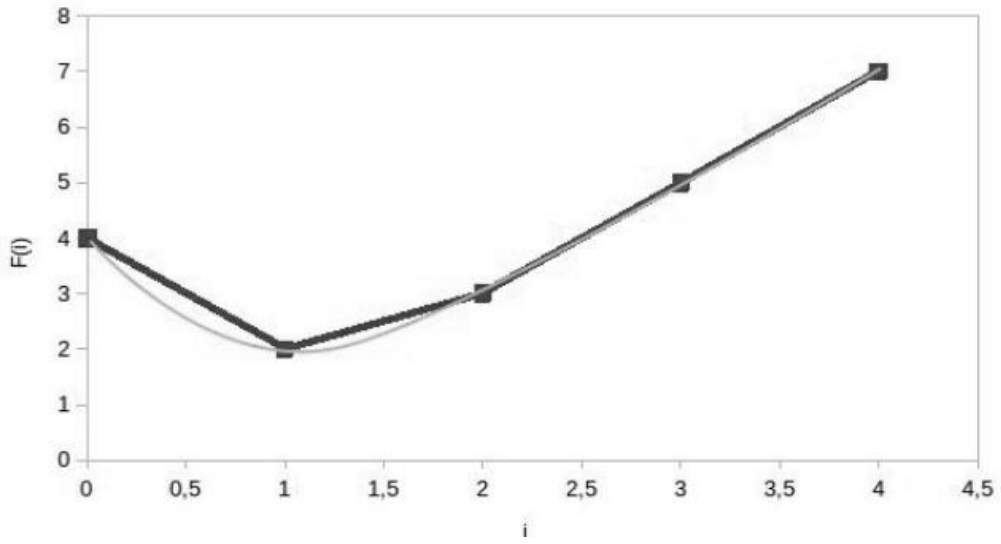


Рисунок 2.3 – Зразок схеми розподілу Шаміра (за [63])

Враховуючи особливості схеми розподілу Шаміра, підсумовуючи, розглянемо переваги та недоліки вказаного методу:

Перевагами схеми розподілу Шаміра можна вважати [64]:

1. Масштабованість. Кількість учасників можна збільшити до поля p, при цьому розмір коаліції, здатної відновити секрет не змінюється.

2. Ідеальність. Розмір кожної з «тіней» дорівнює розміру секрету.

3. Динамічність. Змінюючи багаточлен і перераховуючи «тіні» і зберігаючи секрет незмінним, ймовірність порушення захисту шляхом витоку «тіней» зменшується, так як для отримання секрету потрібно k або більше «тіней», отриманих одному многочлене.

4. Досконалість. Учасники, які володіють разом менш “тінями”, нічого не дізнаються про секрет.

Недоліками схеми Шаміра можна вважати [64]:

1. Дилер-противник. У розглянутій схемі ми враховували можливість, що дилер може видати неправильні проекції.

2. Ненадійність дилера. Дилер може саботувати відновлення секрету.

Схема поділу секрету Шаміра пропонує безпеку з погляду теорії

інформації. Це означає, що математика є стійкою навіть проти зловмисника із необмеженою обчислювальною потужністю. Однак схема, як і раніше, містить кілька відомих проблем.

Наприклад, схема Шаміра не створює фрагментів, що перевіряються, тобто люди можуть вільно пред'являти підроблені фрагменти і заважати відновленню правильного секрету. Ворожий зберігач фрагментів з достатньою інформацією може навіть зробити інший фрагмент, змінивши S на власний розсуд. Ця проблема вирішується за допомогою схем поділу секрету, таких як схема Фельдмана.

Інша проблема полягає в тому, що довжина будь-якого фрагмента дорівнює довжині відповідного секрету, тому довжину секрету легко визначити. Ця проблема вирішується тривіальним набиванням секрету довільними числами до фіксованої довжини.

Нарешті, важливо зазначити, що наші побоювання щодо безпеки можуть виходити за межі самої схеми. Для реальних криптографічних додатків часто існує загроза атак сторонніми каналами, коли зловмисник намагається отримати корисну інформацію з часу виконання програми, кешування, збоїв і т.д. Якщо це викликає занепокоєння, слід під час розробки ретельно розглянути використання захисних заходів, таких як функції та пошук з постійним часом виконання, запобігти збереженню пам'яті на диску та продумати ряд інших речей, які можуть бути опрацьовані в подальших роботах.

2.3 Розробка алгоритму роботи стегосистеми на основі обраних методів

Стеганографічна система або стегосистема – це сукупність засобів та методів, що використовуються для формування прихованого каналу передачі інформації.

При побудові стегосистеми повинні враховуватися такі обов’язкові умови:

– супротивник має повне уявлення про стеганографічну систему та деталі

її реалізації. Єдиною інформацією, яка залишається невідомою потенційному противнику, є ключ, за допомогою якого тільки його власник може встановити факт присутності та зміст прихованого повідомлення;

– якщо противник якимось чином дізнається про факт існування прихованого повідомлення, це не дозволить йому витягти подібні повідомлення в інших даних доти, доки ключ зберігається в таємниці;

– потенційний противник повинен бути позбавлений будь-яких технічних та інших переваг у розпізнаванні чи розкритті змісту таємних повідомлень.

Аналогічно до криптографії, за типом стегоключа стегосистеми можна підрозділити на два типи: з секретним ключем; з відкритим ключем.

У стегосистеми з секретним ключем використовується один ключ, який повинен бути визначений або до початку обміну секретними повідомленнями, або переданий по захищеному каналу.

У стегосистеми з відкритим ключем для вбудовування і вилучення повідомлення використовуються різні ключі, які розрізняються таким чином, що за допомогою обчислень неможливо вивести один ключ з іншого. Тому один ключ (відкритий) може передаватися вільно по незахищеному каналу зв'язку. Крім того, дана схема добре працює і при взаємній недовірі відправника і одержувача.

Узагальнена модель стегосистеми представлена на рис. 2.4.

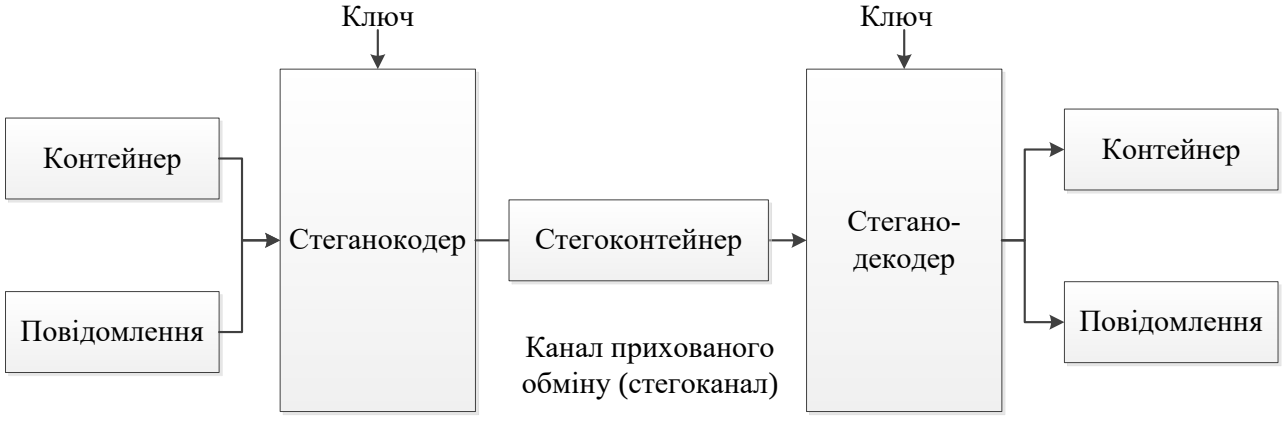


Рисунок 2.4 – Узагальнена модель стегосистеми (за [54 – 55])

В якості даних, в стеганографічній системі можуть використовуватися інформація будь-якого формату: текстова, повідомлення, графічна тощо.

Отже, суть роботи полягає у підвищенні стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

В загальному структура роботи реалізована таким чином:

Крок 1. Користувач обирає зображення (стегоконтейнер для вбудовування даних).

Крок 2. У відповідне поле вписується текст, що повинен бути вкраплений у обране зображення.

Крок 3. Введений для приховування текст являє собою секрет.

В свою чергу, користувачеві необхідно вказати на скільки частин такий секрет потрібно розділити, та скільки таких частин уде потрібно для відновлення даних із обраного стегоконтейнера.

Крок 4. Після реалізації кроку 3, відбувається запуск процесу вбудовування даних, користувачеві необхідно здійснити збереження певної кількості зображень (залежить від кількості частин розподілу секрету). Такі зображення зберігаються у папці result.

Крок 5. Для реалізації процесу вилучення, користувачеві слід завантажити в додаток файли із секретом та запустити процес вилучення.

Крок 6. Якщо користувач вірно завантажив усі частини секрету та здійснив вилучення, приховане повідомлення буде показано користувачеві у відповідному вікні.

Під час здійснення процесів вкраплення і вилучення даних у відповідні зображення, реалізуються згадування вище процеси: схема розподілу Шаміра та НЗБ-метод.

Отже, далі розглянемо більш детальніше процеси вкраплення та вилучення даних, розробимо алгоритм роботи програмного додатку із врахуванням усіх особливостей поставленої задачі.

Алгоритм роботи програмного додатку для реалізації можливості

вкраплення даних.

Крок 1. Запуск виконуваного файлу додатку.

Крок 2. Вибір розділу додатку «вкраплення даних».

Крок 3. Вибір зображення – стегоконтейнера для вкраплення даних.

Крок 4. Перевірка відповідності параметрів обраного файлу встановленим вимогам (формат, розмір, кольоровий режим).

Крок 4.1. У випадку, якщо параметри завантаженого файла не відповідають встановленим вимогам – користувач отримує про це відповідне повідомлення та повторно виконує вибір файлу – стегонтейнеру.

Крок 4.2. У випадку, якщо параметри завантаженого файла відповідають встановленим вимогам – даний етап вважається пройденим та відбувається перехід до кроку 5.

Крок 5. Введення даних для вкраплення. Дані для вкраплення можуть бути завантажені користувачем у вигляді текстового документа, або введенні у вигляді тексту у відповідне поле.

Якщо користувач завантажує текстовий файл із даними – система автоматично перевіряє відповідність вимогам параметрів даного файлу.

Крок 5.1. У випадку, якщо параметри завантаженого файла не відповідають встановленим вимогам – користувач отримує про це відповідне повідомлення та повторно виконує вибір файлу – даних.

Крок 5.2. У випадку, якщо параметри завантаженого файла відповідають встановленим вимогам – даний етап вважається пройденим та відбувається перехід до кроку 6.

Крок 6. Задання необхідної кількості сторін для розподілу даних.

Крок 7. Задання необхідної кількості сторін достатніх для відновлення даних.

Крок 8. Підтвердження процесу вкраплення.

Крок 9. У випадку якщо процес вкраплення успішний – користувачеві необхідно зберегти окремі файлі із вбудованими даними. Якщо процес вкраплення був невдалим – відкривається відповідне повідомлення, а процес

вбудовування слід проводити повторно.

Крок 10. Завершення роботи з програмою на етапі вкраплення.

Схематично алгоритм роботи програмного додатку для реалізації можливості вкраплення даних наведений на рисунку 2.5.

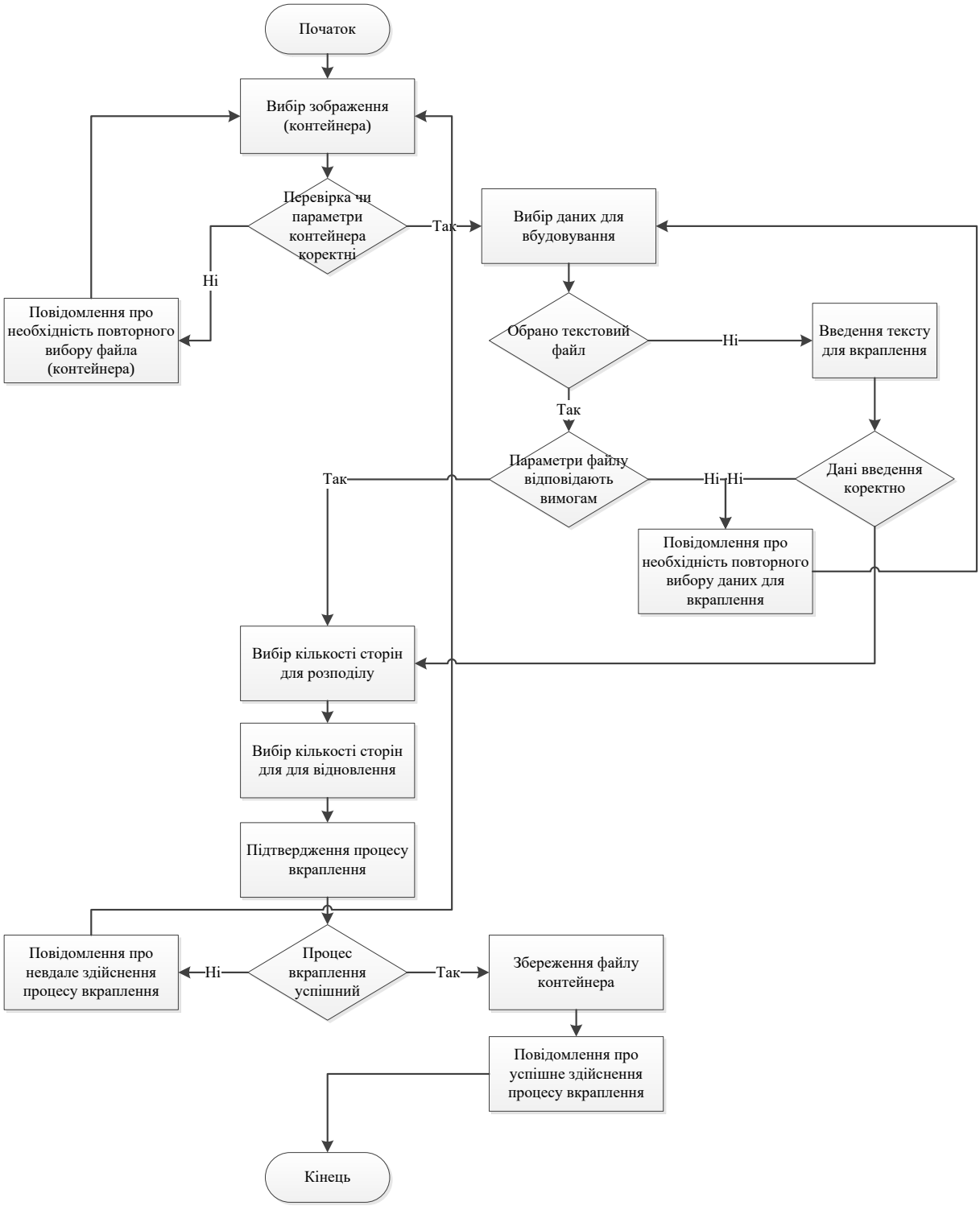


Рисунок 2.5 – Алгоритм роботи програми для здійснення процесу вкраплення

Алгоритм роботи програмного додатку для реалізації можливості вилучення даних.

Крок 1. Запуск виконуваного файлу додатку.

Крок 2. Вибір розділу додатку «вилучення даних».

Крок 3. Вибір файлу-контейнера, що є однією із частин розподіленого секрету.

Крок 4. Перевірка відповідності параметрів обраного зображення заданим вимогам.

Крок 4.1. У випадку, якщо обране зображення не є стегоконтейнером та не розпізнане системою, користувач отримує відповідне повідомлення та існує необхідність повторного здійснення даного процесу.

Крок 4.2. У випадку, якщо обране зображення коректне, після підтвердження користувачем запуск процесу вилучення, у відповідному полі діалогового вікна додатку з’являється частина секрету, яку потрібно перенести у вкладку «відновлення даних».

Такі дії потрібно виконати для усієї кількості зображень, що містять розподілений секрет. Варто зауважити, що кількість цих зображень задається власне, самим користувачем при вкраплені даних у відповідний контейнер (зображення).

Крок 5. Після того, як користувач вилучив секрет із необхідної кількості зображень, усі частини секрету повинні бути перенесені у вкладку «відновлення даних». Кількість частин секрету повинна відповідати вказаній мінімальній кількості при вкраплені даних.

Крок 6. Після підтвердження запуску процесу відновлення даних, користувач отримує відповідне повідомлення.

Крок 6.1. У випадку, якщо усі секрети були вилучені та задані коректно, користувачеві відкривається вихідне приховане повідомлення.

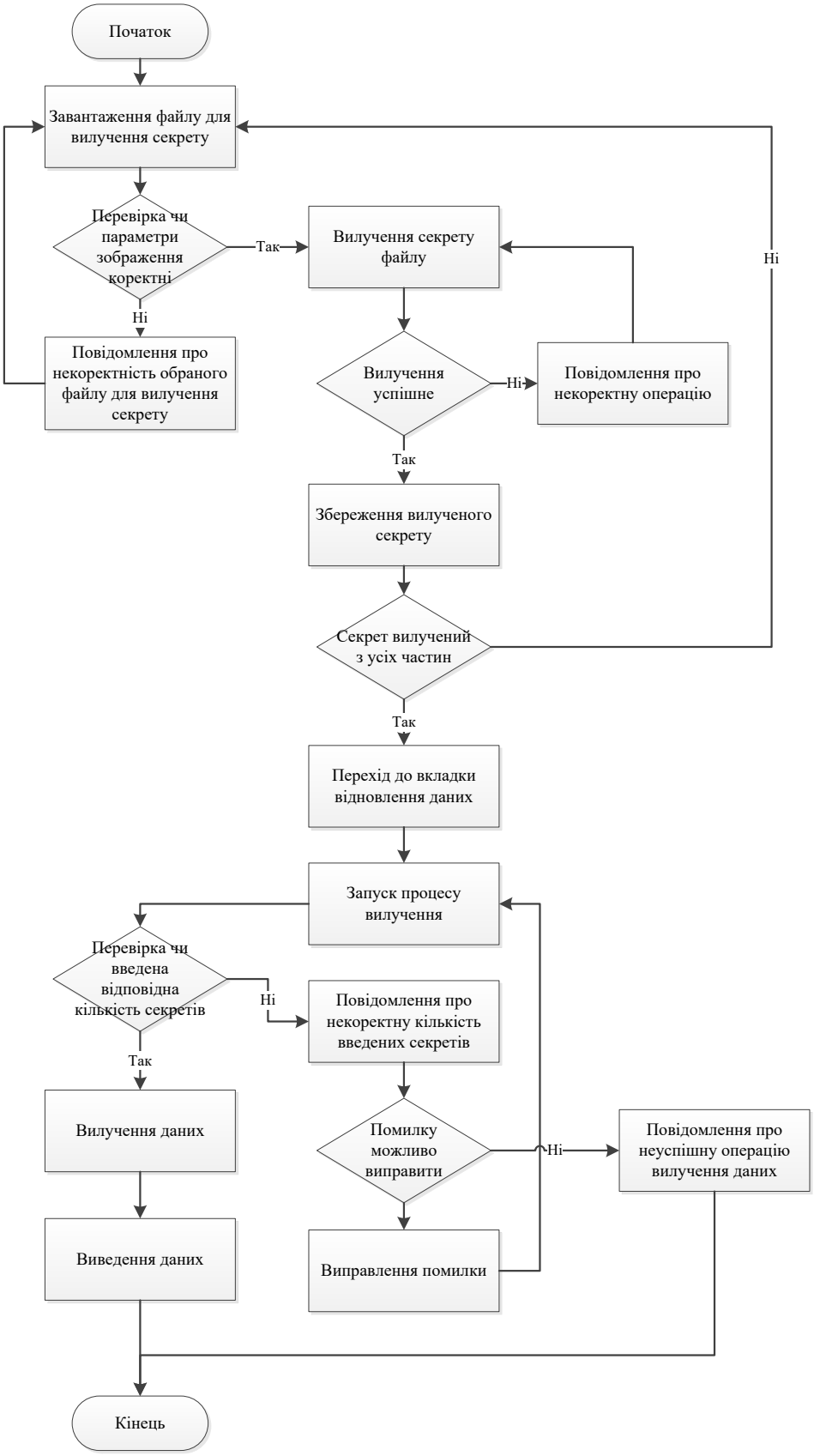


Рисунок 2.6 – Алгоритм роботи програми для здійснення процесу вилучення даних

Крок 6.2. У випадку, якщо секрети завантажені невірно, або їх кількість не відповідає заданим параметрам, користувач отримує повідомлення з інформацією про помилку та необхідність її усунення (наприклад, видалення зайвого секрету, або навпаки, додання ще одного).

Якщо помилка, допущена користувачем, виправлена можна здійснювати повторну спробув вилучення даних.

Крок 6.3. У випадку, якщо користувач намагається використати підробні дані, використовує підробні файли – система такі дані не приймає, можливість скоригувати секрети – відсутня. Відповідно, вилучення даних також неможливе.

Після вилучення даних із файлу контейнера користувач може продовжити роботу з додатком, записати нове повідомлення і т.д. У випадку, якщо необхідності подальшої роботи з додатком у даний час немає, робо а в програмі на такому етапі може бути завершена.

Таким чином, у даному підрозділі було здійснено розробку алгоритму роботи програмного засобу для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

2.4 Обґрунтування вибору мови та засобу програмування

Враховуючи поставлені задачі роботи, для програмної реалізації додатку для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра доцільним є використання мови об’єктно-орієнтованого програмування C# у середовищі Visual Studio.

– C# створювалася як мова компонентного програмування, і в цьому одне з головних переваг мови, спрямоване на можливість повторного використання створених компонентів. З інших об'єктивних факторів відзначимо наступні:

– C# створювався паралельно з каркасом Framework .Net і повною мірою враховує всі його можливості – як FCL, так й CLR;

– C# є повністю об'єктно-орієнтованою мовою, де навіть типи, вбудовані в мову, представлені класами;

– C# є потужною об'єктною мовою з можливостями спадкування й універсалізації;

– C# є спадкоємцем мов C/C++, зберігаючи кращі риси цих популярних мов програмування;

– завдяки каркасу Framework .Net, що стали надбудовою над операційною системою, програмісти C# одержують ті ж переваги роботи з віртуальною машиною, що й програмісти Java. Ефективність коду навіть підвищується, оскільки виконавче середовище CLR являє собою компілятор проміжної мови, у той час як віртуальна Java-машина є інтерпретатором байта-коду;

– потужна бібліотека каркасів підтримує зручність побудови різних типів додатків на C#, дозволяючи легко будувати Web-служби, інші види компонентів, досить просто зберігати й одержувати інформацію з бази даних й інших сховищ даних;

– реалізація, що сполучає побудову надійного й ефективного коду, є немаловажним чинником, що сприяє успіху C#.

Ім'я .Net. Імена нинішнього покоління продуктів від Microsoft супроводжуються закінченням .Net (читається Dot Net), що відбиває бачення Microsoft сучасного комунікативного світу. Комп'ютерні мережі поєднують людей і техніку. Людина, що працює з комп’ютером або використовує мобільний телефон, природно стає частиною локальної або глобальної мережі. У цій мережі використаються різні спеціальні пристрої, починаючи від космічних станцій і кінчаючи датчиками, розташованими, наприклад, у готелях і що посилають інформації про об'єкт всім мобільним пристроям у їхній околиці. У глобальному інформаційному світі комунікативна складова будь-яких програмних продуктів починає відігравати визначальну роль.

У програмних продуктах .Net за цим ім'ям стоїть цілком конкретний зміст, що припускає, зокрема, наявність відкритих стандартів комунікації, перехід від створення автономних додатків до створення компонентів, що

допускають повторне використання в різних середовищах і додатках. Можливість повторного використання вже створених компонентів і легкість розширення їхньої функціональності - все це неодмінні атрибути нових технологій. Важливу роль у цих технологіях грає мова XML, що стала стандартом обміну повідомленнями в мережі.

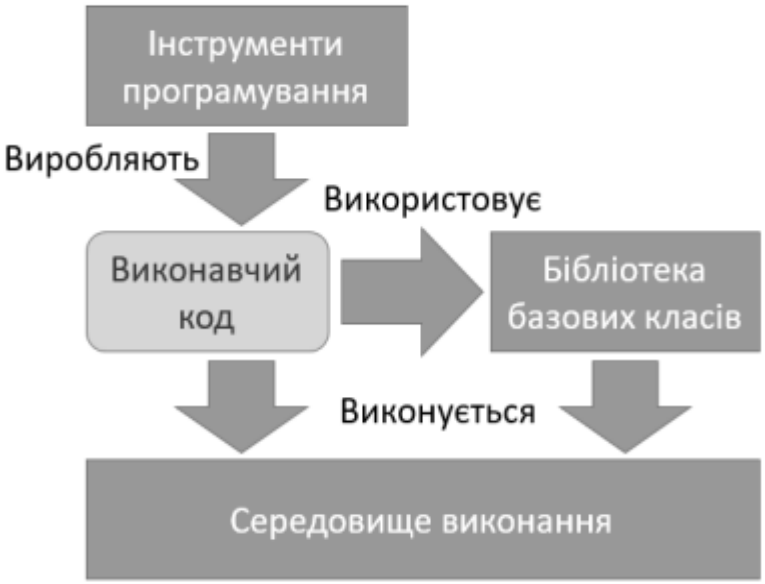


Рисунок 2.7 – Компіляція .NET фреймворк (за [65])

Не намагаючись охопити все різноманіття мережної взаємодії, розглянемо реалізацію нових ідей на прикладі Visual Studio .Net. .

Visual Studio .Net – відкрите середовище розробки [66].

Середовище розробки Visual Studio .Net – це вже перевірений часом програмний продукт, що є сьомою версією Студії. Але новинки цієї версії, пов'язані з ідеєю .Net, дозволяють вважати її принципово новою розробкою, що визначає новий етап у створенні програмних продуктів. Виділю дві найважливіші, на мій погляд, ідеї:

– відкритість для мов програмування;

– принципово новий підхід до побудови каркасу середовища – Framework .Net.

Середовище розробки тепер є відкритим мовним середовищем. Це означає, що поряд з мовами програмування, включеними в середовище фірмою

Microsoft - Visual C++ .Net (з керованими розширеннями), Visual C# .Net, J# .Net, Visual Basic .Net, - у середовище можуть додаватися будь-які мови програмування, компілятори яких створюються іншими фірмами-виробниками. Таких розширень середовища Visual Studio зроблено вже досить багато, практично вони існують для всіх відомих мов - Fortran й Cobol, RPG й Component Pascal, Oberon й SmallTalk.

Відкритість середовища не означає повної волі. Всі розроблювачі компіляторів, при включенні нової мови в середовище розробки, повинні додержуватися певних обмежень. Головне обмеження, яке можна вважати й головним досягненням, полягає в тому, що всі мови, що включають у середовище розробки Visual Studio .Net, повинні використати єдиний каркас - Framework .Net. Завдяки цьому досягаються багато бажаних властивостей: легкість використання компонентів, розроблених на різних мовах; можливість розробки декількох частин одного додатка на різних мовах; можливість написати клас на одній мові, а його нащадків - на інших мовах. Єдиний каркас приводить до зближення мов програмування, дозволяючи разом з тим зберігати їх індивідуальність і наявні в них переваги. Завдяки єдиному каркасу, Visual Studio .Net у певній мірі вирішує це завдання у світі програмістів.

Рисунок 2.8 – Компіляція вихідного коду у збірку (за [67])

Framework .Net – єдиний каркас середовища розробки

У каркасі Framework .Net можна виділити два основних компоненти:

– статичний - FCL (Framework Class Library) - бібліотеку класів каркаса;

– динамічний – CLR (Common Language Runtime) – загальномовне виконавче середовище [67].

Таким чином, в результаті використання розробленого алгоритму та з допомогою обраних інструментів реалізації,далі в роботі заплановано здійснити практичну реалізацію додатку для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

2.5 Висновки до розділу

Отже, в даному розділі здійснювалась робота над розробкою та проектуванням програмного додатку для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

В ході роботи було досліджено особливості алгоритм роботи методу найменш значущого біта та схеми розподілу секрету Шаміра, розроблено алгоритму роботи стегосистеми на основі обраних методів (алгоритм роботи програми для здійснення процесу вкраплення, алгоритм роботи програми для здійснення процесу вилучення даних).

Із врахуванням поставлених задач та особливостей розробки для її програмної реалізації було обґрунтовано вибір мови та середовища програмування.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКУ ПІДВИЩЕННЯ СТІЙКОСТІ ТА ПРОПУСКНОЇ ЗДАТНОСТІ ЦИФРОВИХ ГРАФІЧНИХ КОНТЕЙНЕРІВ

3.1 Проектування графічного користувацького інтерфейсу

Принципи розробки інтерфейсу – це концепції та уявлення високого рівня, які можна використовувати під час проектування програмного забезпечення [68].

Необхідно визначити, який із принципів найбільш важливий і прийнятний для вашої системи.

Три принципи розробки користувальницького інтерфейсу формулюються так:

1. Контроль за користувачем інтерфейсу;

2. Зменшення завантаження пам'яті користувача;

3. Послідовність користувача інтерфейсу.

На сьогоднішній день ці принципи не зовсім підходять для графічних інтерфейсів, проте, в найближчому майбутньому вони безперечно будуть застосовуватися.

Принципи створення сумісного інтерфейсу:

1. Проектування послідовного інтерфейсу.

При переміщенні в інтерфейсі користувачі повинні мати опорні точки. Це заголовки вікон, навігації, карти та деревоподібні структури. Також повинна бути можливість завершити поставлену задачу без зміни середовища роботи або переключення між стилями введення інформації.

2. Загальна сумісність усіх програм.

Одним із головних аспектів у розробці інтерфейсу є можливість навчання користувача концепціям системи та програмного продукту, що застосовуються в нових ситуаціях та в інших програмах. Прояв місткості можна відстежити на трьох рівнях:

– подача інформації (мається на увазі, що користувачі можуть сприймати інформацію та об'єкти в подібному логічному, візуальному та фізичному вигляді у всьому програмному продукті;

– поведінка програми;

– техніка взаємодії.

3. Збереження результатів взаємодії.

Перед виконанням дії необхідно інформувати користувача, якщо результати можуть бути відмінні від тих, що він очікує. Наділіть його опціями здійснення дій, можливості їх скасування або відтворення.

4. Естетична привабливість та цілісність.

Недолік у функціональності програмного продукту не повинен бути прихований приємним для погляду інтерфейсом.

5. Заохочення вивчення.

Створення дружнього інтерфейсу є однією з важливих завдань проектувальників користувальницьких інтерфейсів. Він заохочував би користувачів на дослідженні. Варто помітити та враховувати при розробці програмного продукту, що користувачі під час його використання користування чекають на допомогу, напрямки, інформацію і, навіть, розваги.

Сучасні інтерфейси – інтуїтивні, передбачувані, дружні і привабливі. Зростання попиту на CD-ROM продукти, нашестя браузерів Internet, домашніх сторінок та прикладних програм подарувало користувачам комп'ютерів цілий світ.

Зараз дружні інтерфейси перетворюються у зручні та що залучають у використанні навіть у програмах, які призначені для організації та ведення бізнесу.

Далі розглянемо особливості розробки інтерфейсу діалогових вікон для розроблюваного додатку для Підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

Програмний додаток не розрахований на велику кількість вікон і, загалом, вся робота користувача зосереджена у одному діалоговому вікні, що містить декілька розділів.

У лівій частині такого діалогового вікна планується розміщення зображень (контейнерів для вбудовування інформації, або вже оброблених зображень, що міститимуть частину секрету). Також заплановано розміщення кнопки «Обрати файл» для звернення до системи ПК і обрання відповідного файлу, а також поля для відображення шляху завантаження обраного файлу.

Права частина діалогового вікна має дещо більше наповнення та поділена на три розділи.

Перший розділ призначений для вкраплення даних у зображення. У даній частині діалогового вікна користувачу надається можливість додати текстовий файл із необхідною інформацією ля вкраплення, ввести текст у відповідне поле, також орати кількість сторін для розподілу та кількість сторін, що буде достатньо для відновлення вкрапленої інформації.

Відповідно, у нижньому правому кутку заплановано розташування кнопки «Вкрапити» для підтвердження процесу вкраплення даних у обране зображення.

Другий розділ призначений для вилучення даних із стегоконтейнера. У відповідному полі відображається секрет вилучений із зображення.

Відповідно, у нижньому правому кутку заплановано розташування кнопки «Вилучити» для підтвердження процесу вилучення даних із обраного зображення.

Третій розділ призначений для відновлення даних з використанням секретів,які були попередньо вилучені та знаходяться у відповідному полі даного розділу.

Відповідно, у нижньому правому кутку заплановано розташування кнопки «Відновити» для підтвердження процесу відновлення даних на основі вилучених секретів.

Спроектований вигляд описаного діалогового вікна наведений на рис. 3.1

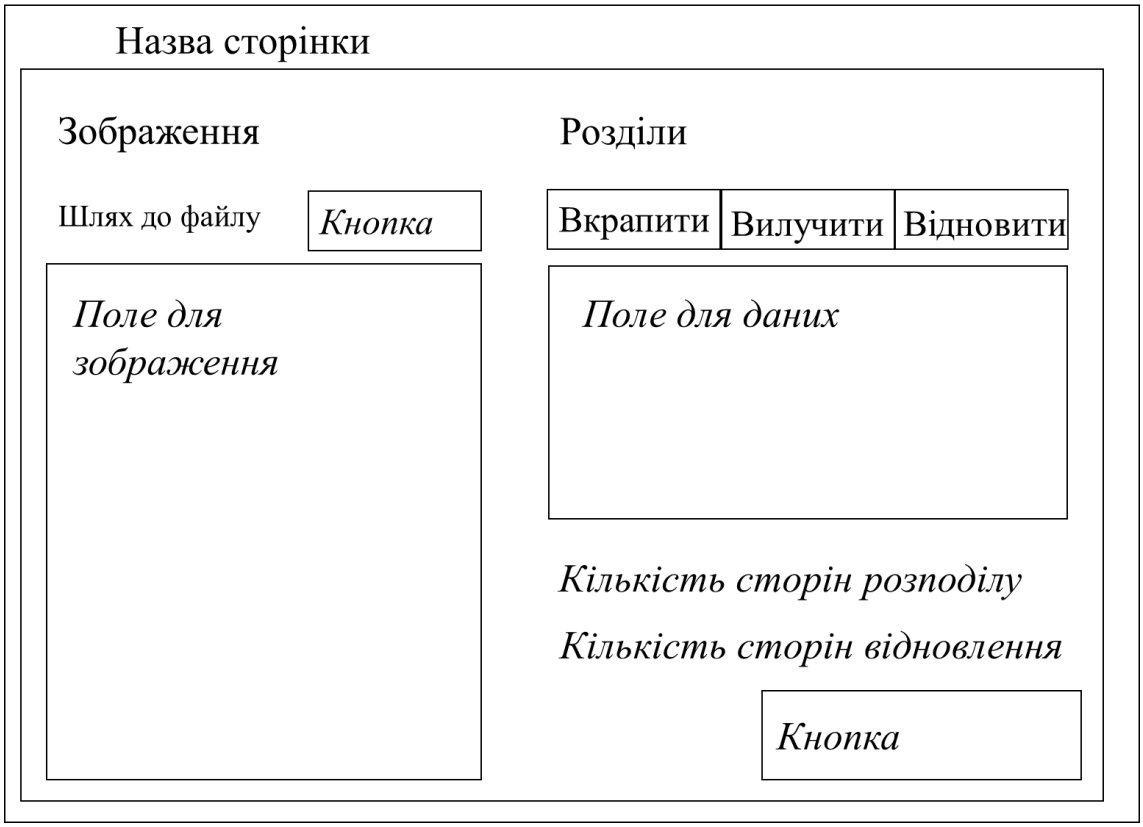


Рисунок 3.1 – Проектування головного діалогового вікна додатку

Таким чином, на основі проаналізованих правил та принципів розробки графічного користувацького інтерфейсу та визначених задач розроблюваного програмного додатку, було спроектованого головне діалогове вікно програмного засобу, де заплановано практично реалізувати поставлену мету роботи.

3.2 Програмна реалізація додатку

Далі в роботі розглянемо розроблені та застосовані ключові кодові послідовності, на основі яких реалізована програмна реалізація розроблюваного продукту та практичне втілення запропонованого функціоналу додатку.

Бібліотеки, що використовувались.

```
using Moserware.Security.Cryptography;
using System;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Linq;
```

```
using System.Text;
using System.Text.RegularExpressions;
using System.Windows.Forms;
```

Введення секрету для вбудовування або додавання файлу з відповідним текстом.

```
public partial class Form1 : Form
  {
     public const string DialogFilter = "Split Secret Files (*.splitsecret)|*.splitsecret|GNU Privacy Guard Files (*.gpg)|*.gpg|All Files (*.*)|*.*";
     public const string AllFilesFilter = "Text files (*.txt)|*.txt";
     private bool _HasAgreedToSafetyWarning;
     private Control _LastSelectedShareTypeControl;
```

Кнопка «Вкрапити», повідомлення про необхідність обрати зображення, якщо даний крок не виконаний користувачем.

```
 private void button1_Click(object sender, EventArgs e)
     {
        if (image == null )
        {
           MessageBox.Show("Для вбудовування необхідно обрати зображення");
           return;
        }
        if (rdoHaveSecretMessage.Checked)
        {
           var shares = CreateSecretMessageShares();
           if (!String.IsNullOrEmpty(shares))
           {
              txtShares.Text = shares;
           }
        }
        else if (rdoHaveSecretFile.Checked)
        {
           CreateSecretFileShares();
        }

     }
```

Шифрування файлу за заданим ключем.

```
private void CreateSecretFileShares()
     {
        if (!TryVerifyThresholdValues())
        {
           return;
        }
        if (!File.Exists(txtSecretFilePath.Text))
```

```
            {
               ShowError("Secret file does not exist", "File Not Found");
               return;
            }
            if (String.IsNullOrEmpty(txtMasterKey.Text))
            {
               ShowError("Для шифрування файлу потрібен дійсний ключ.", "Invalid Key
Length");
               return;
            }
```

Повідомлення про невдале шифрування у зв'язку з некоректним ключем.

```
byte[] keyBytes;
            if (!SecretEncoder.TryParseHexString(txtMasterKey.Text, out keyBytes))
            {
               ShowError("The key must contain all hexadecimal characters.", "Invalid Key
Character");
               return;
            }
```

Розподіл секрету.

```
var sb = new StringBuilder();
            sb.AppendLine("Here are your secret pieces that form the decryption key for the
encrypted file located at:");
            sb.AppendLine(saveFileDialog.FileName);
            sb.AppendLine();
            int shareWidth = saveFileDialog.FileName.Length;
            foreach (var currentShare in splitSecret.GetShares((int)nudShares.Value))
            {
               string currentShareText = currentShare.ToString();
               sb.AppendLine(currentShareText);
               shareWidth = Math.Max(shareWidth, currentShareText.Length);
               sb.AppendLine();
            }
```

Визначення розміщення частин секрету.

```
private void SetWidthFromShareLength(int shareWidth)
         {
            var textSize = TextRenderer.MeasureText(new string('0', shareWidth),
txtShares.Font);
            // HACK to get shares to to not wordwrap
            var calculatedWidthFromText = textSize.Width + 70;
            var screenBounds = Screen.FromControl(txtShares).Bounds;
            var screenLeft = Left - screenBounds.Left;
            var screenWidth = screenBounds.Width;
            Width = Math.Max(Math.Max(Math.Min(calculatedWidthFromText, screenWidth -
screenLeft), MinimumSize.Width), Width);
         }
```

Процес вбудовування секрету:

```
private string CreateSecretMessageShares()
{
   if (!TryVerifyThresholdValues())
   {
      return null;
   }
   if (String.IsNullOrWhiteSpace(txtSecretMessage.Text))
   {
      ShowError("Enter something for the secret message", "Empty secret");
      return null;
   }
   var sb = new StringBuilder();
   sb.AppendFormat("Your secret message has been split into the following {0} pieces:", nudShares.Value);
   sb.AppendLine();
   sb.AppendLine();
   int maxShareWidth = 0;
   int index = 1;
   var extension = textBox1.Text.Split('.')[1];
   var extensionFile = "." + extension;
   saveFileDialog.FileName = index + extensionFile;
   DialogResult res = saveFileDialog.ShowDialog();
   if (res == System.Windows.Forms.DialogResult.OK)
   {
      foreach (var currentSplit in SecretSplitter.SplitMessage(txtSecretMessage.Text,
         (int)nudThreshold.Value, (int)nudShares.Value))
      {
         sb.AppendLine(currentSplit);
         maxShareWidth = Math.Max(maxShareWidth, currentSplit.Length);
         sb.AppendLine();
         Embed(currentSplit, index, extensionFile);
         index++;
      }
   }
```

Повідомлення про успішне вбудовування секрету.

```
MessageBox.Show("Вбудовування успішно завершено");
   SetWidthFromShareLength(maxShareWidth);
   sb.AppendFormat("To reconstruct your secret, you'll need to provide exactly {0} of the above pieces. Please remember to keep the pieces safe and give them only to people you trust.", nudThreshold.Value);
   sb.AppendLine();
```

Обробка сторони розподілу.

```
private void Embed(string text, int index, string extensionFile)
   {
```

```
            Bitmap encrypted = Steganography.Steganography.InsertEncryptedTextToImage(pictureBox1.Image, text);
            if (encrypted != null)
            {
               var path = Path.GetDirectoryName(saveFileDialog.FileName);
               var dir = path + "\\result";
               if (!(Directory.Exists(dir)))
               {
                  Directory.CreateDirectory(dir);
               }
               encrypted.Save($"{dir}\\output{index}{extensionFile}");
            }
         }
```

Якщо частина із секретом обрана невірно.

```
 {
            if (nudThreshold.Value <= nudShares.Value)
            {
               return true;
            }

            ShowError("The number of required pieces must equal or exceed the number of pieces.", "Invalid minimum piece count");
            return false;
         }
```

Попередження про коректний вибір частин із розподіленим секретом.

```
      private void SetRecoverTextToExample()
         {
            inputText.Text = @"Майте на увазі, що ви повинні ввести лише мінімальну кількість секретних частин, необхідних для відновлення секрету.
      Наприклад, лише 2 із наведених нижче секретних фрагментів необхідні для відновлення зразка секретного повідомлення:
```

Повідомлення про необхідність введення коректних даних для вилучення.

```
      [SAMPLE SECRET]
      Видаліть один з наведених вище рядків і натисніть ""Recover Secret""";
            inputText.Text = inputText.Text.Replace("[SAMPLE SECRET]", CreateSecretMessageShares());
         }
```

Якщо секрет у обраному зображенні не знайдено.

```
      private void btnRecover_Click_1(object sender, EventArgs e)
         {
            if (String.IsNullOrWhiteSpace(inputText.Text))
            {
               if (MessageBox.Show(this, "No secret pieces were specified. Would you like to see
```

```
an example?", "Example?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.Yes)
                {
                  SetRecoverTextToExample();
                }
                return;
              }
```

Помилка під час дешифрування.

```
        catch (ModificationDetectedException modificationDetectedException)
                {
                  ShowError("It looks like the file was tampered with or the given secret pieces
were invalid.",
                        "Modification Detected");
                }
                catch (Exception exception)
                {
                  ShowError("Під час спроби розшифрувати файл сталася помилка. Будь
ласка, переконайтеся, що ви ввели лише найменшу кількість необхідних секретних частин,
вибрали правильний файл для розшифрування та чи файл не був підроблений.");
                }
              }
```

Некоректність дешифрованих даних.

```
        else
                {
                  txtSecretMessage.Text = combinedSecret.RecoveredHexString;
                  ShowMessage(MessageBoxIcon.Warning, "Відновлене повідомлення містить
символи, які не можна друкувати. Зазвичай це означає, що ви вказали занадто багато або
занадто мало секретних частин. Якщо ви вважаєте, що у вас занадто багато секретних частин,
спробуйте видалити одну і подивіться, чи це допоможе. В іншому випадку спробуйте додати
ще один секретний елемент. (Показ декодованого повідомлення як двійкове значення)»,
«Виявлено символи, які не можна друкувати");
                }
```

Вибір файлу із секретними даними.

```
        private void btnBrowsePlaintext_Click(object sender, EventArgs e)
            {
              openFileDialog.Filter = AllFilesFilter;
              openFileDialog.Title = "Виберіть файл, що містить секретну інформацію";
              openFileDialog.CheckFileExists = true;
              if (openFileDialog.ShowDialog() == DialogResult.OK)
              {
                txtSecretFilePath.Text = openFileDialog.FileName;
              }
            }
```

Кнопка відновлення секрету.

```
private void button3_Click(object sender, EventArgs e)
    {

        openFileDialog.FileName = "*.*";
        DialogResult res = openFileDialog.ShowDialog();
        if (res == System.Windows.Forms.DialogResult.OK)
        {
            if (image != null)
            {
                image.Dispose();
            }
            string ext = Path.GetExtension(openFileDialog.FileName);
            if (ext == ".png" || ext == ".bmp" || ext == ".jpg")
            {
                try
                {
                    image = Image.FromFile(openFileDialog.FileName);
                    textBox1.Text = openFileDialog.FileName;
                    pictureBox1.Image = image;
                    currentMode = Mode.Image;
                    audio = null;
                    // audioLabel.Visible = false;
                }
                catch
                {
                    image = null;
                    pictureBox1.Image = null;
                }
            }
        }
    }
```

Повідомлення про успішне вилучення секрету.

```
private void decryptButton_Click_1(object sender, EventArgs e)
    {
        if (currentMode == Mode.Image && image != null)
        {
            string text = Steganography.Steganography.GetDecryptedTextFromImage(image);

            if (text != null)
            {
                textBox3.Text = text;
                MessageBox.Show("Секрет успішно вилучено");
            }
```

Повідомлення про помилку під час вилучення секрету.

```
else
        {
          MessageBox.Show("Це зображення не містить секрету або сталася помилка");
        }
```

Таким чином, з використанням можливостей обраних засобів програмування, програмних методів та класів було реалізовано програмний засіб для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра із врахуванням усіх поставлених умов та задач роботи.

3.3 Реалізація користувацького інтерфейсу

Під час реалізації користувацького інтерфейсу використовувались спроектовані на початку розділу, зразки діалогових вікон. Розглянемо покроково діалогові вікна програми під час роботи з додатком.

Після запуску виконуваного файлу додатку, відкривається головне діалогове вікно. У лівій частині даного вікна розташована кнопка «Обрати», поле для відображення шляху до файлу та поле, де буде розміщено обраний файл.

У такому полі розміщення файлу відображається зображення, що піддається обробці (контейнер в який вже вудований секрет, або який тільки підлягає вбудовуванню).

У правій частині вікна розташовано три розділи, що відповідають трьом етап роботі з додатком «вкрапити дані», «вилучити секрет», «відновити дані».

Нижче розташоване поле для вибору тектового файлу з даними для вкраплення і поле для введення даних для вкраплення.

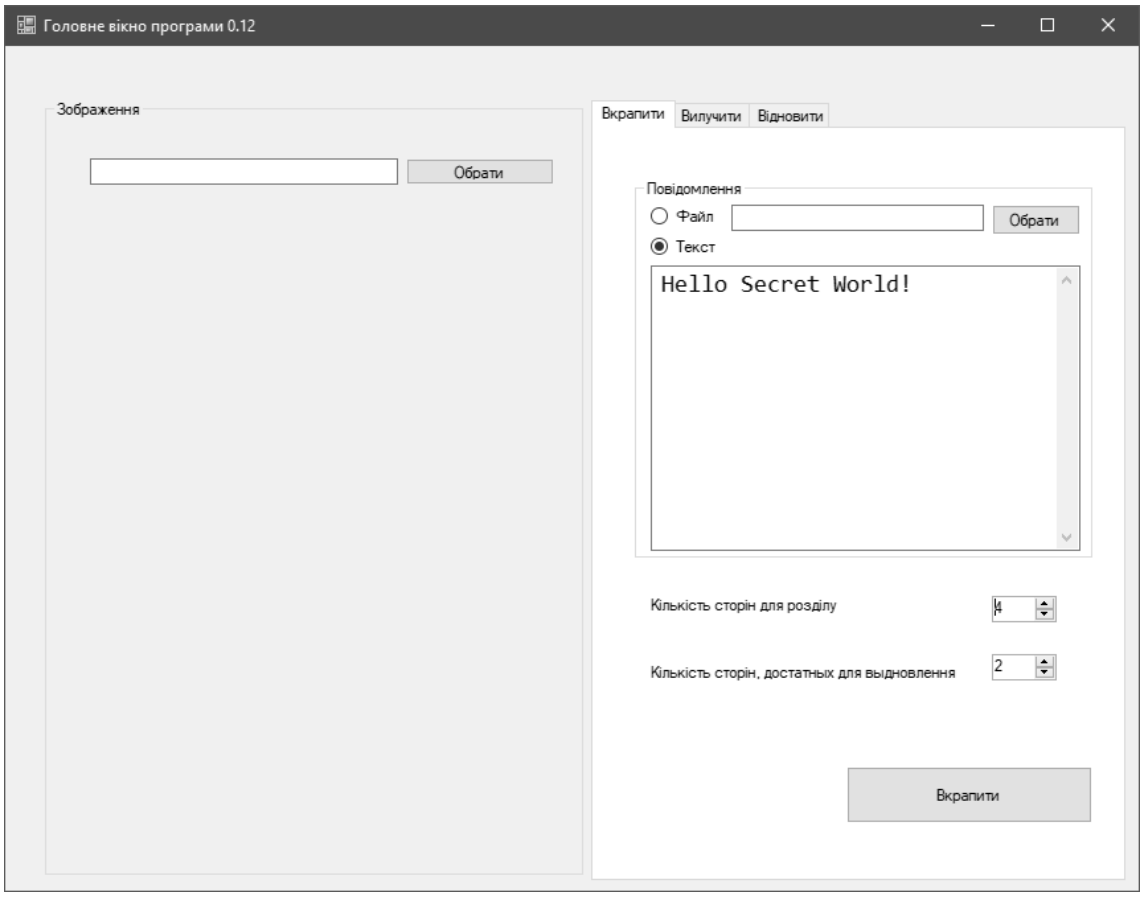


Рисунок 3.2 – Вигляд головного діалогового вікна додатку

Ще однією важливою частиною даного діалогового вікна є поля для визначення кількості сторін для розподілу та кількості сторін достатніх для відновлення.

Відповідно, нижче розташовується кнопка «Вкрапити» для запуску відповідного процесу.

Далі, на рисунку 3.3 наведено зразок діалогово вікна із заповненими полями файла-стегоконтейнера та введеними даними для вкраплення.

У випадку, якщо усі дані введені коректні і процес вбудовування даних успішний, користувач отримує про це відповідне повідомлення (рис. 3.4).

Результатом успішного вбудовування є зображення із вкрапленим секретом, відповідну кількість таких зображень користувач зберігає для подальшої передачі.

Зображення зберігаються під своїми порядковими номерами у папці «result».

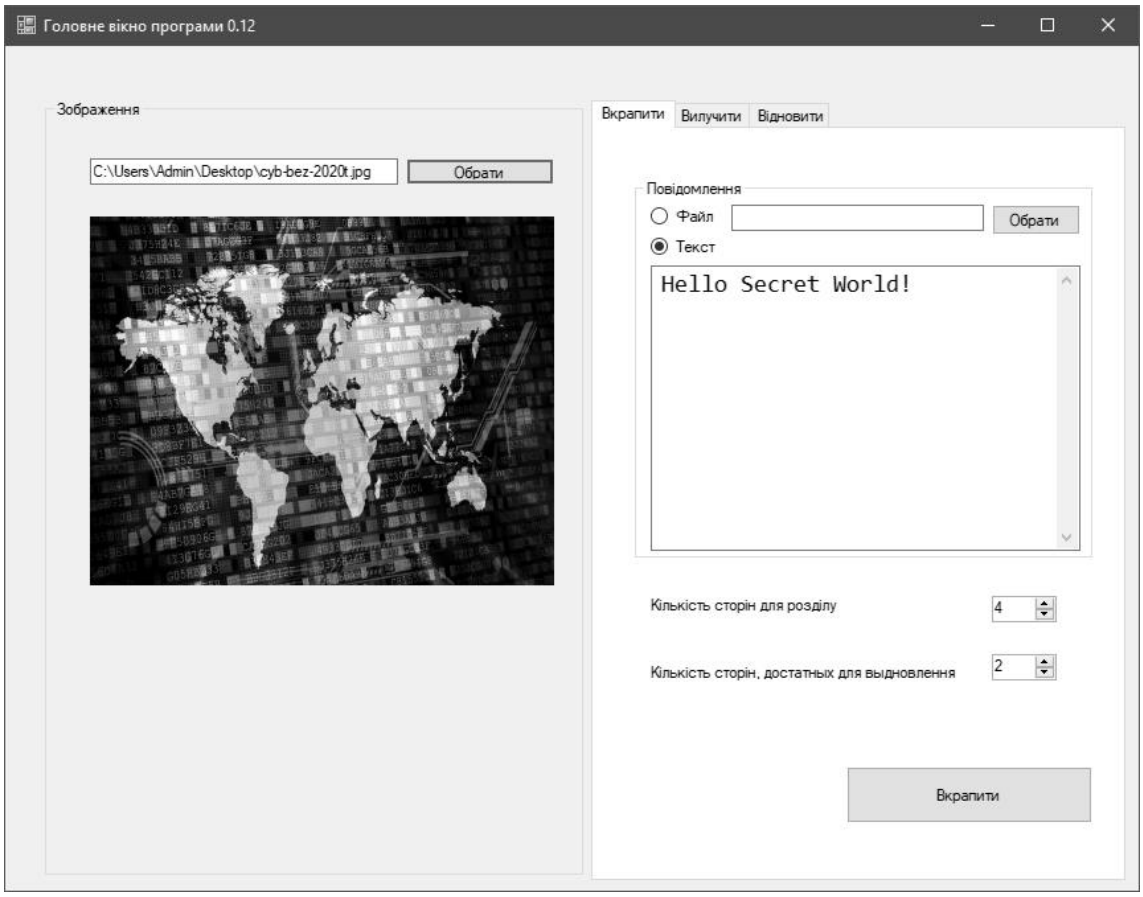


Рисунок 3.3 – Вигляд діалогового вікна із заповненими вхідними даними

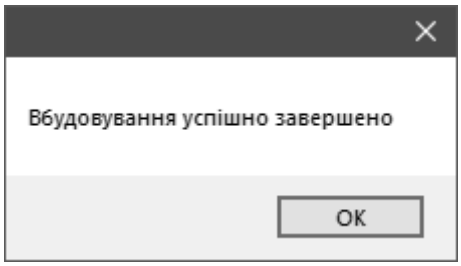


Рисунок 3.4 – Повідомлення про успішне здійснення процесу вбудовування

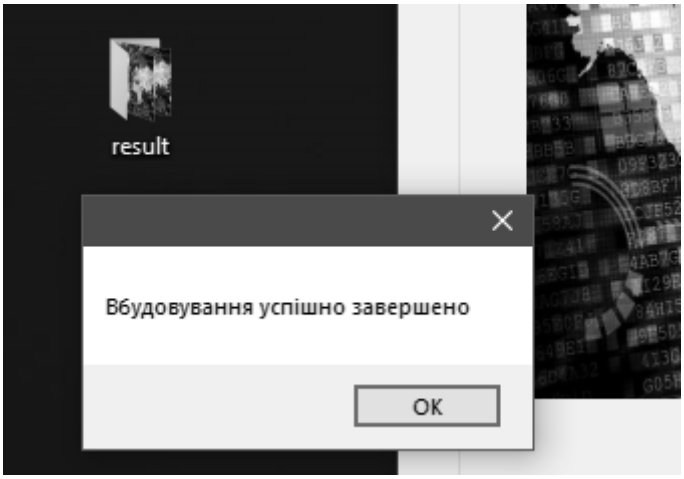


Рисунок 3.4 – Збереження даних вкраплення даних

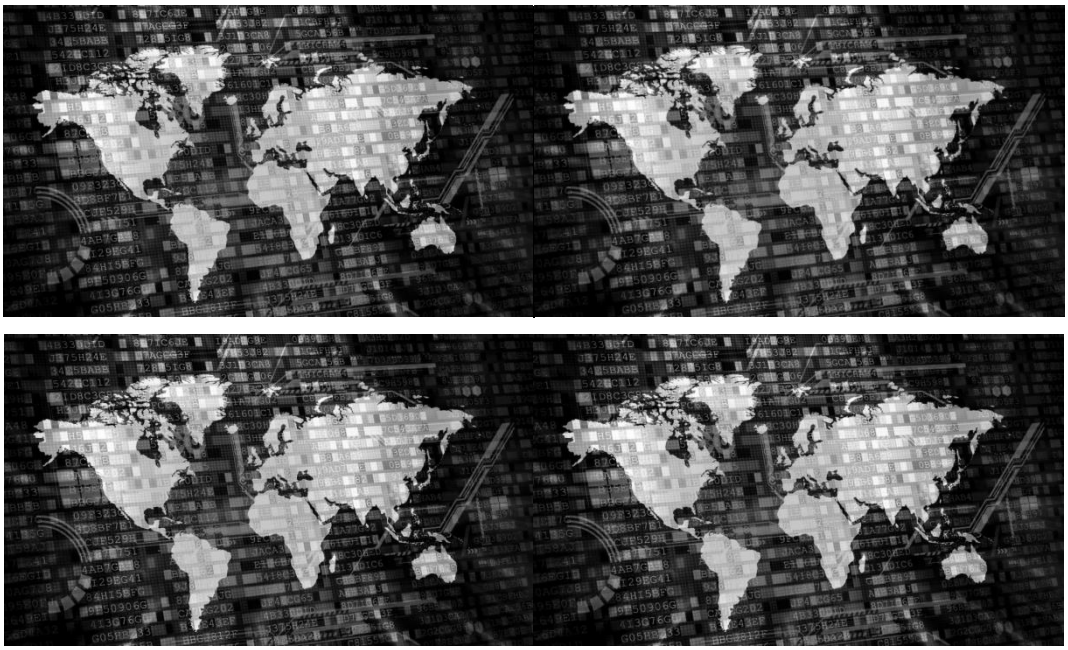


Рисунок 3.5 – Зображення-контейнери, як результат вкраплення даних

На рисунку 3.6 наведено приклад вигляду збереження оброблених файлів із вкрапленими даними.

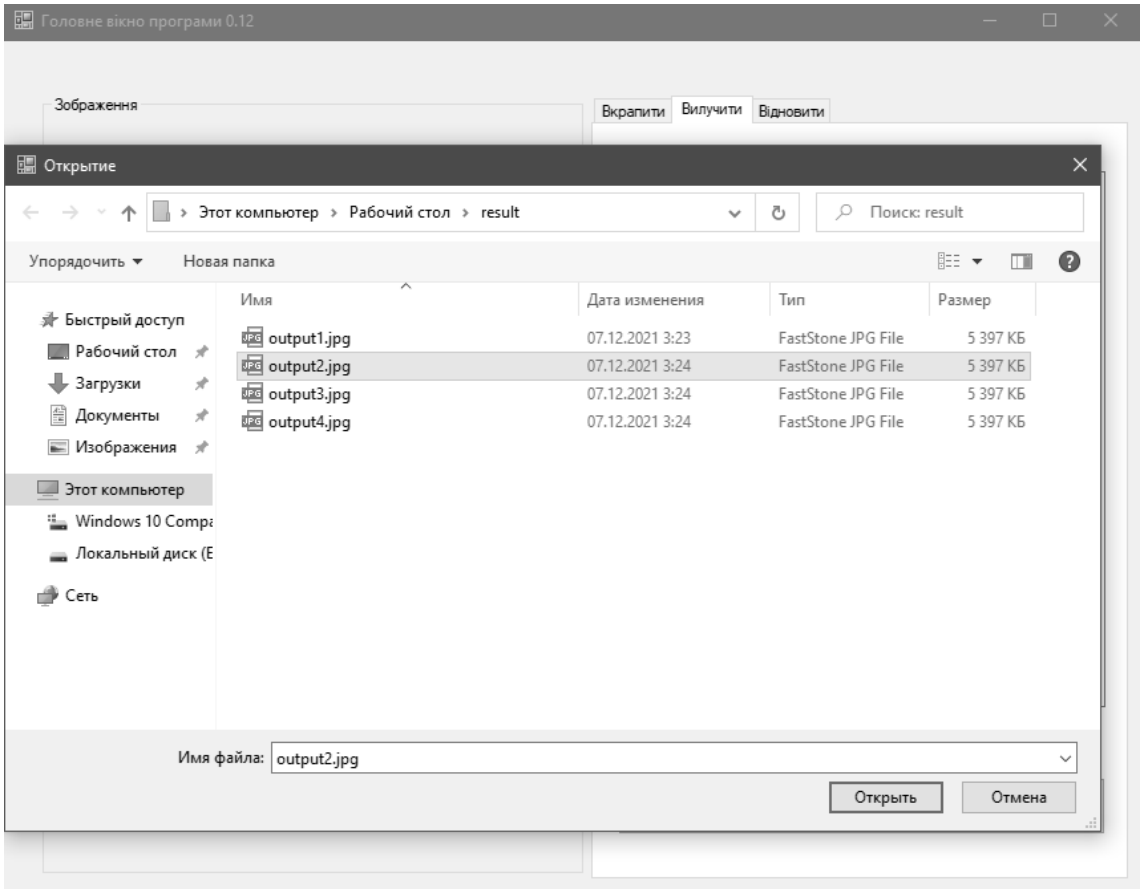


Рисунок 3.6 – Приклад збереження файлів із вкрапленими даними

Для вилучення даних із наявних зображень, користувачеві необхідно перейти у вкладку «вилучення даних», у лівій частині вікна обрати відповідний файл для вилучення та підтвердити процес відповідною кнопкою.

У випадку, якщо файл із вбудованими даними обрано коректно, система розпізнає секрет та виведе його для користувача у відповідне поле. Корисутвачеві про такий результат буде повідомлено у окремому діалоговому вікні.

Після отримання першої частини секрету, користувачеві необхідно його скопіювати та перенести у вкладку «відновлення даних» для подальшої роботи з ним. Вигляд даного діалогового вікна наведено на рис. 3.8.

Оскільки після того, як будуть вилучені усі неохідні секрети – користуваче слід самостійно здіймнити процес відновлення даних.

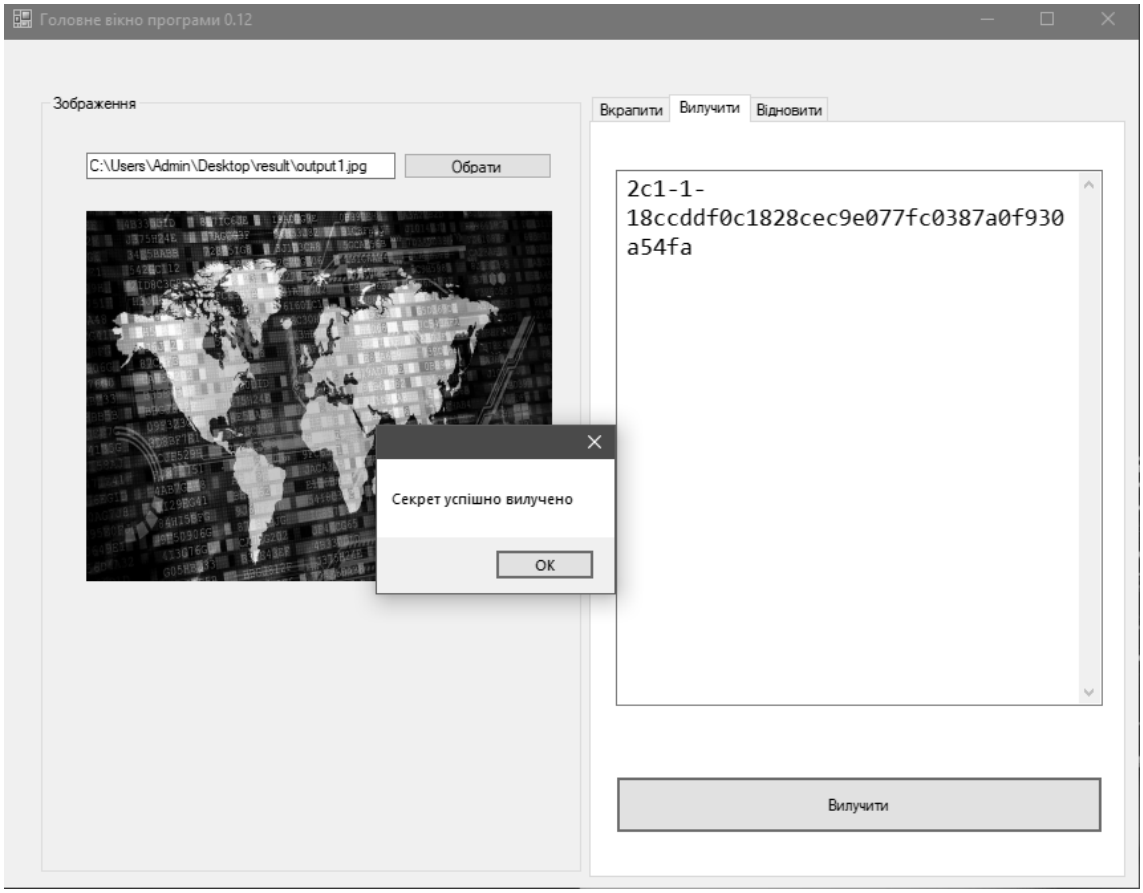


Рисунок 3.7 – Вигляд діалогово вікна при здійсненні процесу вилучення секрету

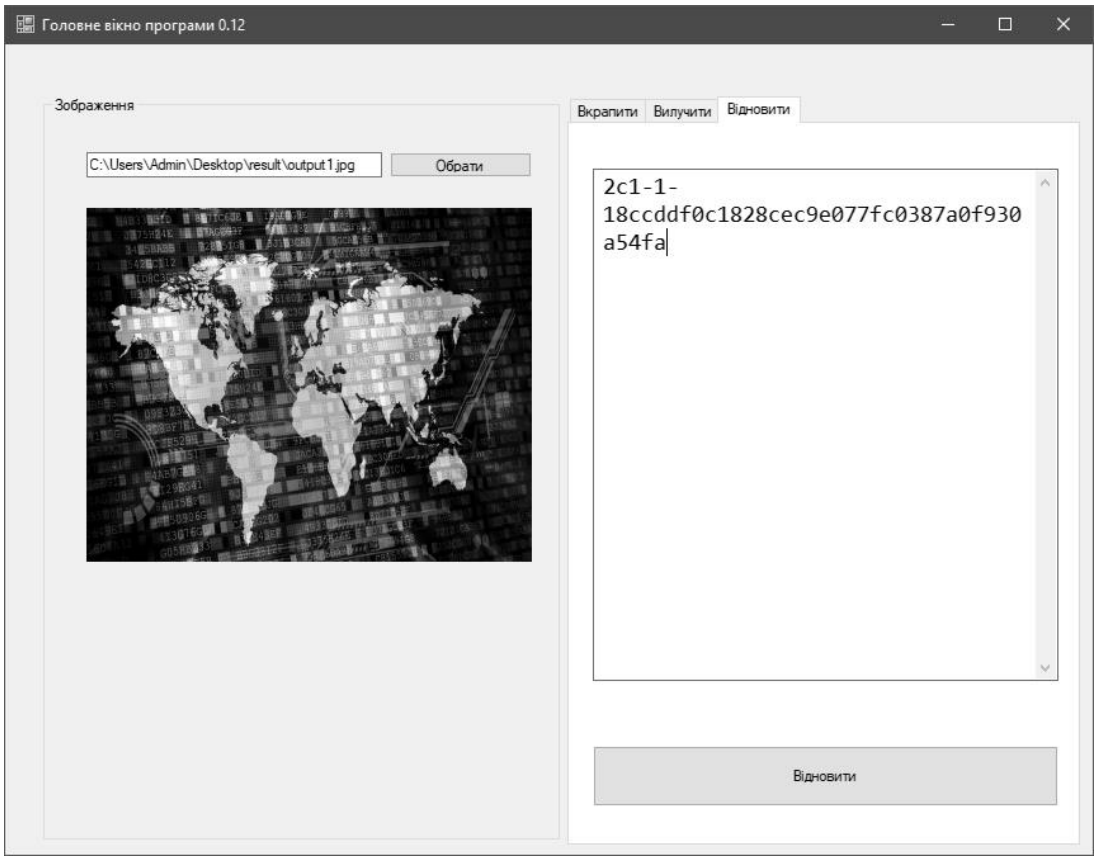


Рисунок 3.8 – Вигляд діалогово вікна вкладки «відновлення даних»

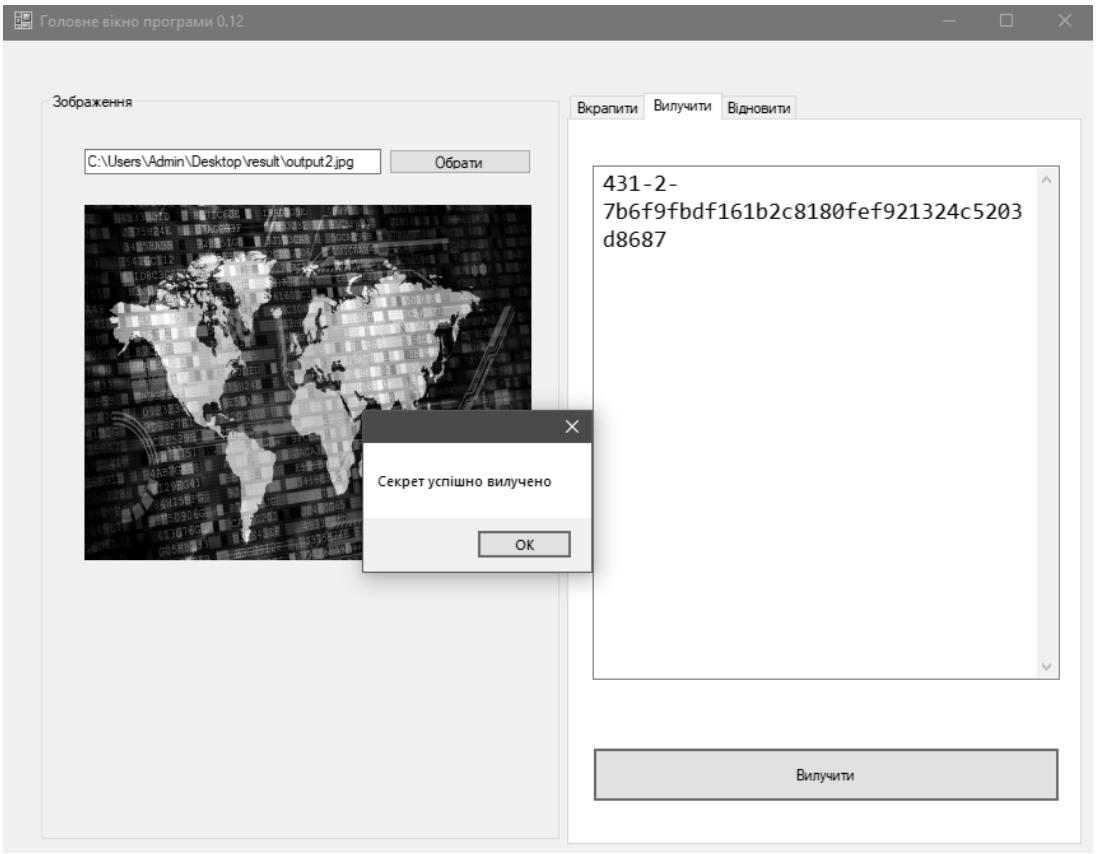


Рисунок 3.9 – Вигляд вікна успішного вилучення другої частини секрету

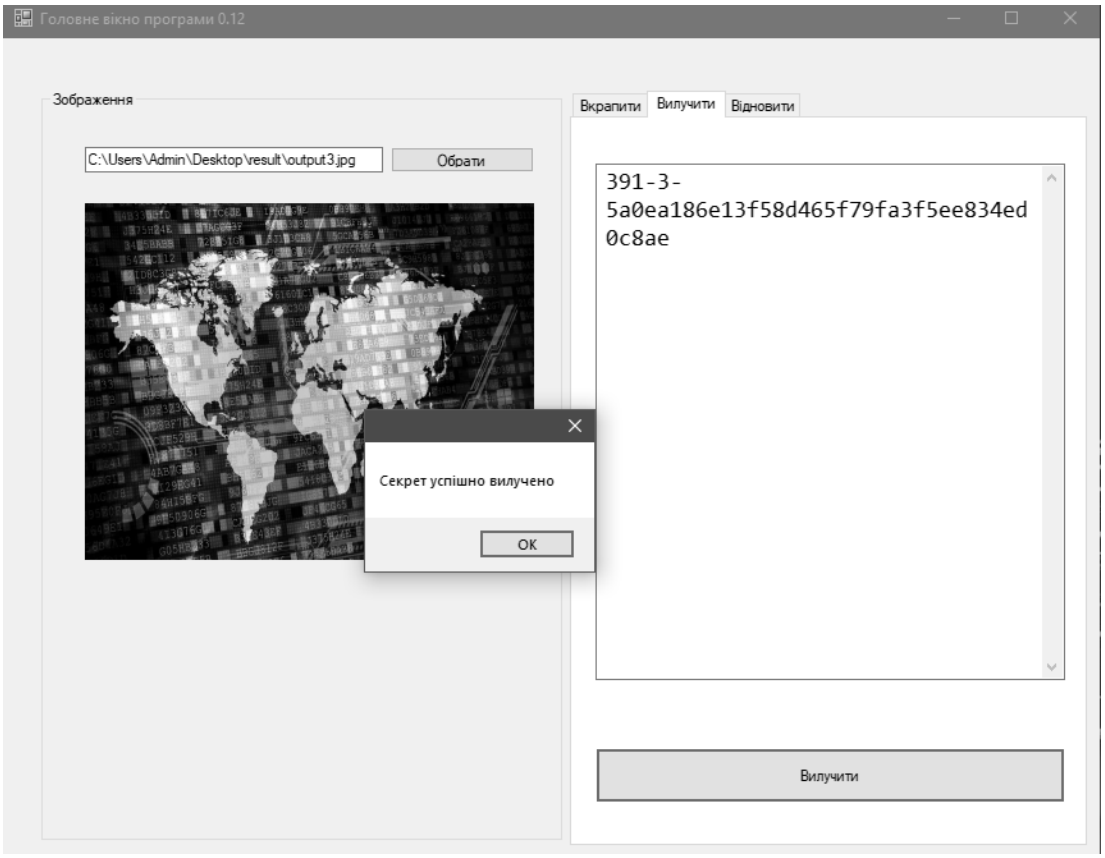


Рисунок 3.10 – Вигляд вікна успішного вилучення третьої частини секрету

Як згадувалося раніше, усі частини відновлених секретів користувачеві необхідно перенести у вкладку «відновлення даних».

Після того, як усі секрети переміщені у відповідне поле, користувачеві слід підтвердити процес запуску відновлення даних натиснувши кнопку «Відновити» (рис. 3.11).

У випадку, якщо введена користувачем кількість секретів не відповідає умовам заданим при вбудовуванні даних, користувач отримає про це відповідне повідомлення (рис. 3.12).

Натиснувши кнопку погодження з помилкою, користувач має змогу її усунути (наприклад, видалити зайвий секрет, або здійснити вилучення секрету із відповідного зображення якщо цього не було зроблено раніше).

Якщо користувач дотримується інструкції та коректно виконує вказані дії, то процес відновлення матиме успішний результат, у відповідному діалоговому вікна користувачеві відкриється відновлене повідомлення (рис. 3.13).

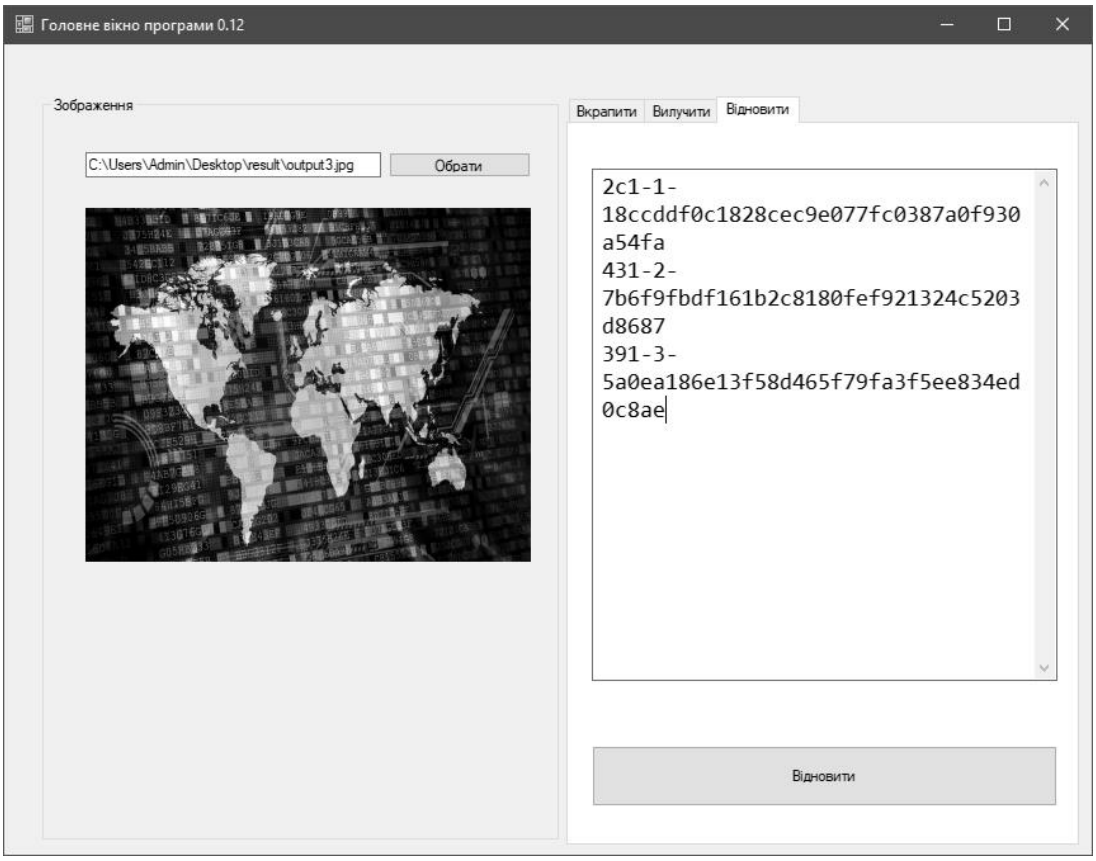


Рисунок 3.11 – Вигляд діалогового вікна вкладки «відновлення даних»

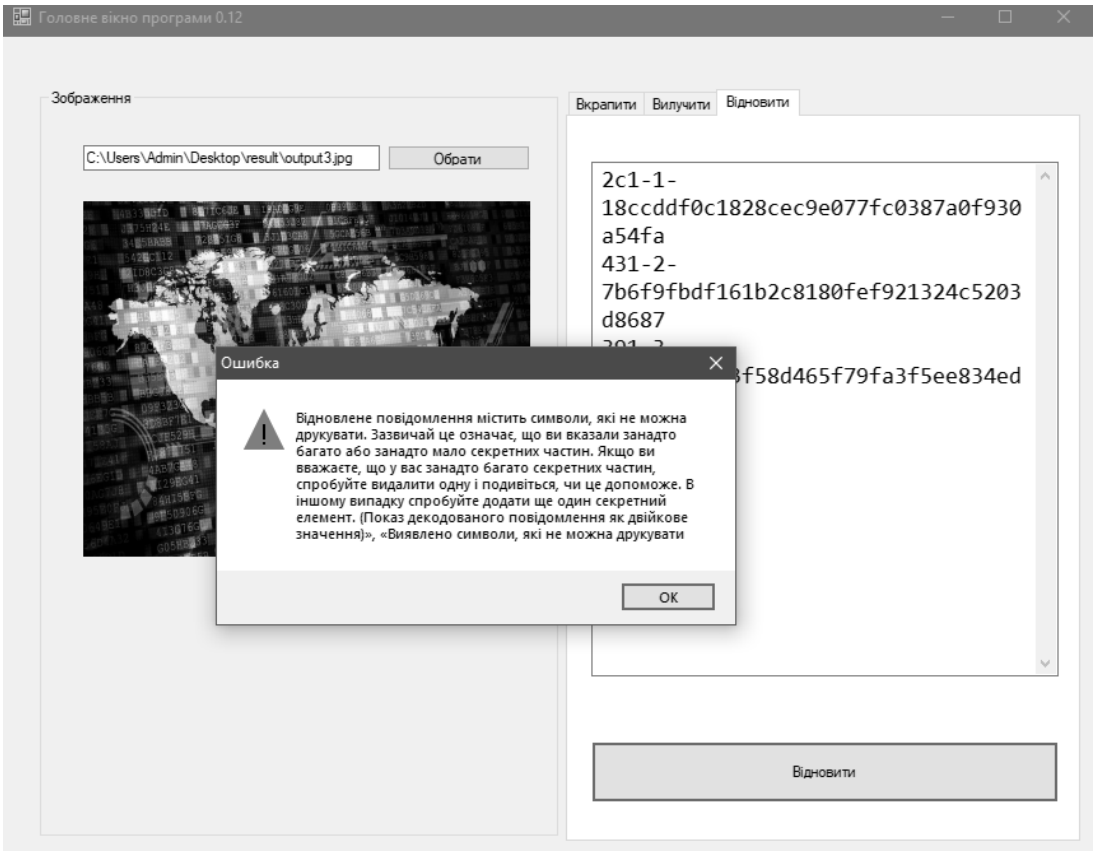


Рисунок 3.13 – Вигляд вікна із повідомленням про помилку відновлення

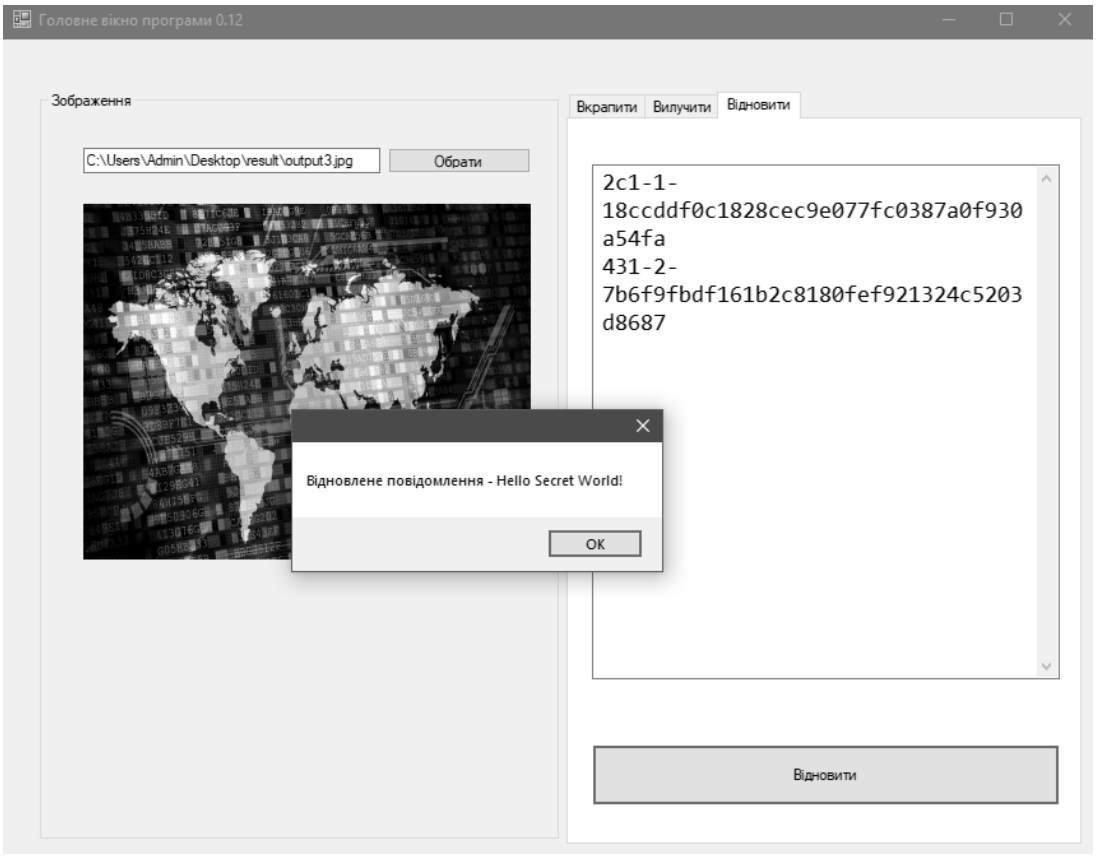


Рисунок 3.14 – Вигляд діалогово вікна у випадку успішного процесу відновлення даних

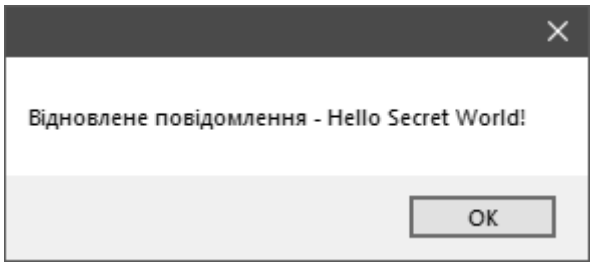


Рисунок 3.15 – Вигляд діалогов вікна із тексто відновленого повідомлення

Таким чином, у результаті здійсненої роботи, було реалізовано практично програмний додаток для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра із наведеної покроковою інструкцією користувача та детальним описом процесів,що реалізовані в додатку.

3.4 Тестування елементів розробленого програмного додатку

У межах роботи, було розроблено модифікований метод існуючого алгоритму найменш значущого біту для вкраплення інформації в контейнер у поєднанні із схемою розподілу Шаміра. Завдяки модифікації, алгоритм коректно працює з повідомленнями, які вкраплюються, підвищуючи стійкість та пропускну здатність. Хоча алгоритм не став ідеальним для передачі даних шляхом вкраплення їх у контейнер, проте не з занадто великими повідомленнями він працює ідеально.

Оскільки вкраплення у випадковому порядку за ключем здійснюється не послідовно з певним кроком, де простежується закономірність, а у непередбачуваному порядку, то це ускладнює завдання аналітикам для виявлення прихованої інформації в контейнері та її коректного вилучення зловмисниками.

Для перевірки практичної доцільності та ефективності реалізованої розробки, здійснимо перевірку практичних результатів з допомогою RS-атаки на стеганоконтейнери із вкрапленими даними.

Отже, одним із оригінальних методів статистичного стегоаналізу є метод RS, вперше опублікованим в 2001 колективом вчених під керівництвом Дж. Фрідріх.

Скорочення в назві розшифровується як RegularSingular, тобто «регулярно-сингулярний» [69].

Суть методу полягає в наступному. Все зображення розбивається на групи по n пікселів G(x,x2,…,xn), де n парно, наприклад, по 2 пікселі, що знаходяться поруч по горизонталі. Для групи пікселів визначається функція регулярності або «гладкості» f(G), як таку функцію можна вибрати, наприклад, дисперсію значень усередині групи, або просто суму перепадів значень суміжних пікселів.

Під значенням піксела розуміємо ціле число від 0 до 255:

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$$

Функція F(x) називається фліпінгом і має властивість F(F(x)) = x. Визначимо дві функції фліппінгу – F1, відповідає інверсії молодшого біта пікселя, і F-1, що є інверсією з перенесенням у старший біт (додавання одиниці):

$$\text{F1: } 0 \leftrightarrow 1,\ 2 \leftrightarrow 3,\ \dots,\ 254 \leftrightarrow 255, \quad (6)$$

$$\text{F-1: } 255 \leftrightarrow 0,\ 1 \leftrightarrow 2,\ 3 \leftrightarrow 4,\ \dots,253 \leftrightarrow 254,\ 255 \leftrightarrow 0.$$

При застосуванні фліпінгу до групи отримуємо перетворену групу пікселів. Далі, поділимо всі групи пікселів на класи таким чином:

$$\text{Регулярні групи: } G\ R \Leftrightarrow f(F(G)) > f(G);$$

$$\text{Сингулярні групи: } G\ S \Leftrightarrow f(F(G)) < f(G);$$

$$\text{Групи, що не використовуються: } G\ U \Leftrightarrow f(F(G)) = f(G).$$

Надалі нас цікавитиме співвідношення між RM, SM, UM, R-M, S-M, UM, де індекси M та –M означають відповідно застосування та для отримання розподілу. Мета – визначити як використання повідомлення методом LSB впливатиме на вищеописану статистику груп пікселів.

Метод ґрунтується на статистичному припущенні, що для природного зображення, тобто незаповненого контейнера, характерно наступне:

$$RM \cong R - M \text{ та } SM \cong S - M.$$

Отже, в таблиці 3.1 наведено результати здійснення атак на 50 контейнерів, в які було вкраплено інформацію звичайним методом НЗБ та модифікованим методом НЗБ у поєднанні із схемою розподілу Шаміра.

В таблиці вказано точність атак на контейнери, тобто усереднене відношення виявленого обсягу інформації до справжньої.

Таблияця 3.1 – Результати здійсненої RS-атаки на стеганоконтейнери

Наповненість контейнера	Звичайний метод НЗБ	Модифікований метод НЗБ + схема розподілу Шаміра
0	0	0
20	45	8,7
35	67,2	15,5
50	79,8	50,1
60	96,7	74,5
75	98,8	83,2
100	100	100

Збільшилась значно часова складність алгоритму на відновлення початкового повідомлення, оскільки для відновлення застосовується поліном Лагранжа. Це досягнуто завдяки використанню схеми розподілу секрету Шаміра до початкового повідомлення, що вкрай суттєво підвищує стійкість контейнеру, оскільки навіть у разі вилучення інформації з контейнеру, її неможливо самотужки відновити до початкового вигляду без інших частин розподілу, що гарантує метод Шаміра.

Отже, навіть після вилучення, для зловмисника в контейнері не знайдеться ніякої корисної інформації.

Таким чином, можна помітити, що модифікований алгоритм у поєднанні із схемою розподілу Шаміра більш коректно працює з міркувань пропускної здатності ніж звичайний НЗБ алгоритм, важче точно виявити обсяг вкрапленої інформації атаками та відновлення початкового повідомлення навіть після вилучення інформації у правильному порядку з контейнера.

З точки зору апаратного забезпечення, цей метод вимагає непомітно більше швидкості, для сучасних комп’ютерів.

3.5 Висновки до розділу

Отже, в даному розділі здійснювалась практична реалізація програмного додатку для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

В ході роботи було здійснено проектування користувацького інтерфейсу користувача, описано програмну реалізацію, детально описано інструкцію користувача для роботи з додатком.

Тестування практичних результатів показало, що модифікований алгоритм у поєднанні із схемою розподілу Шаміра більш коректно працює з міркувань пропускної здатності ніж звичайний НЗБ алгоритм, важче точно виявити обсяг вкрапленої інформації атаками та відновлення початкового повідомлення навіть після вилучення інформації у правильному порядку з контейнера.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки ПЗ на основі стеганографічних методів

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності [70].

Результатом магістерської кваліфікаційної роботи є розробка програмного засобу для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

Для проведення технологічного аудиту залучено трьох незалежних експертів. У нашому випадку такими експертами є викладачі кафедри МБІС: Карпінець В. В. (к.т.н., доцент каф. МБІС ВНТУ) та Павловський П.В. (викл. каф. МБІС ВНТУ), Шиян А. А. (к.ф.-м.н., доцент каф. МБІС ВНТУ).

Оцінювання комерційного потенціалу було здійснене за критеріями, що наведені в таблиці 4.1

Таблиця 4.1 - Критерії оцінювання комерційного потенціалу розробки бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Крите-р.	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатацій ні витрати значно вищі, ніж в аналогів	Експлуатацій ні витрати дещо вищі, ніж в аналогів	Експлуатацій ні витрати на рівні експл. витрат аналогів	Експлуатацій ні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навч. наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження таблиці 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої к-ті дозвільних документів на вир-во та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки пові-домлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу експертами розробки зведено в таблицю 4.2.

Таблиця 4.2 - Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1 – Карпінець В.В.	2 – Павловський П.В.	3 – Шиян А.А
1	3	4	4
Ринкові переваги (недоліки):			
2	3	4	3
3	4	4	3
4	3	3	4
5	4	4	4
Ринкові перспективи			
6	4	4	3
7	4	3	4
Практична здійсненність			
8	4	3	4
9	3	4	3
10	4	4	3
11	4	3	3
12	3	3	4
Сума балів	$СБ_1 = 43$	$СБ_1 = 43$	$СБ_1 = 42$
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = 42{,}7$		

За даними таблиці 4.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Рівень комерційного потенціалу розробки, становить 42,7 балів, що відповідає рівню «високий».

Проаналізуємо суть технічної проблеми та розглянемо аналоги. Наукова новизна розробки полягає в підвищенні стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

В ході роботи було досліджено особливості алгоритм роботи методу найменш значущого біта та схеми розподілу секрету Шаміра, розроблено алгоритму роботи стегосистеми на основі обраних методів (алгоритм роботи програми для здійснення процесу вкраплення, алгоритм роботи програми для здійснення процесу вилучення даних).

У межах роботи, було розроблено модифікований метод існуючого алгоритму найменш значущого біту для вкраплення інформації в контейнер у поєднанні із схемою розподілу Шаміра.

Завдяки модифікації, алгоритм коректно працює з повідомленнями, які вкраплюються, підвищуючи стійкість та пропускну здатність. Хоча алгоритм не став ідеальним для передачі даних шляхом вкраплення їх у контейнер, проте не з занадто великими повідомленнями він працює ідеально.

Враховуючи такі переваги розробленого методу, можемо порівняти його з основними аналогами.

У таблиці 4.4 наведені основні технічні показники аналога і нового програмного продукту.

Таблиця 4.4 – Основні технічні показники аналога і нового програмного продукту

Показники, %	Аналог	Нова розробка	Відношення параметрів нової розробки до параметрів аналога
Функціональність	80	100	1,25
Надійність	75	100	1,33
Сумісність	95	100	1,05
Супровід	70	100	1,43
Економія ресурсів і часу	90	100	1,11
Простота використання	80	100	1,25

Отже, суть роботи полягає у підвищенні стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

В загальному структура роботи реалізована таким чином: Користувач обирає зображення (стегоконтейнер для вбудовування даних). У відповідне поле вписується текст, що повинен бути вкраплений у обране зображення. Введений для приховування текст являє собою секрет. В свою чергу, користувачеві необхідно вказати на скільки частин такий секрет потрібно розділити, та скільки таких частин уде потрібно для відновлення даних із обраного стегоконтейнера. Після реалізації кроку 3, відбувається запуск процесу вбудовування даних, користувачеві необхідно здійснити збереження певної кількості зображень (залежить від кількості частин розподілу секрету). Такі зображення зберігаються у папці result. Для реалізації процесу вилучення, користувачеві слід завантажити в додаток файли із секретом та запустити процес вилучення. Якщо користувач вірно завантажив усі частини секрету та

здійснив вилучення, приховане повідомлення буде показано користувачеві у відповідному вікні. На підставі вищевикладеного можна стверджувати, що нове технічне рішення, що пропонується для розробки, буде мати кращі показники, ніж у аналога та більшою мірою задовольнить потреби споживачів. Тому його розробка та впровадження є актуальним та доцільним.

Програмний засіб на сьогодні має перспективу та користь як для пересічних користувачів так і для спецслужб. Продукт, який пропонується є реалізованим засобом, що дозволяє проводити автентифікацію користувачів в системі. Готовий програмний продукт буде реалізовуватись на ринку програмних засобів шляхом щомісячної підписки за певну плату.

Під час встановлення ціни та попиту на новий програмний продукт основна увага повинна акцентуватися на унікальності об'єкта купівлі-продажу, цінах продуктів конкурентів, перевагах порівняно з аналогами, витратах, які зазнає покупець у разі заміни старого продукту новим, ступені терміновості та гостроті потреби.

Програмний засіб готовий для використання. Фахівці відповідної кваліфікації наявні, трудові та фінансові ресурси теж, обслуговування програми може відбуватись в режимі он-лайн, з будь-якої точки світу, оскільки немає проблем з передачею на нього прав. Комерціалізація розробки знаходиться на початковому етапі. Ведуться пошуки інвесторів та партнерів. Наявні зацікавлені особи, що готові першими випробувати програмний засіб в обмін на акт впровадження та подальшу рекламу від їх імені. Просування на ринок планується шляхом реалізації та продажу через спеціалізовані магазини програмного забезпечення.

4.2 Прогнозування витрат на виконання наукової роботи та впровадження її результатів

Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи складається з таких етапів:

1-й етап: розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;

2-й етап: розрахунок загальних витрат на виконання даної роботи;

3-й етап: прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Виконаємо розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи, за такими статтями та формулами, приймаючи до уваги те, що для розробки інформаційної технології було залучено одного розробника програмного забезпечення.

1. Основна заробітна $З_о$:

$$З_о = \frac{М}{Т_р} \cdot t, \text{грн.} \quad (4.1)$$

де М – місячний посадовий оклад – 20 000 грн.;

$Т_р$ – число робочих днів в місяці; приблизно $Т_р = 20$ днів;

t – число робочих днів роботи – 30 днів.

Таким чином:

$$З_о = \frac{20000}{20} \cdot 30 = 30\,000 \text{ (грн.)}$$

Таблиця 4.5 – Витрати по заробітній платі

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату
Розробник	20 000	1000	30	30 000
Всього				30 000

2. Додаткова заробітна плата $З_д$ працівників розраховується як 12% від основної заробітної плати:

$$З_д = 0{,}12 \cdot 30\,000 = 3\,600 \text{(грн.)} - \text{для розробника}$$

3. Нарахування на заробітну плату $Н_{зп}$ розробника становить:

$$Н_{зп} = (З_о + З_д) \cdot \frac{\beta}{100} \quad (4.2)$$

де $З_о$ – основна заробітна плата розробника;

$З_д$ – додаткова заробітна плата розробника;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування – 22%.

$$Н_{зп} = (30\,000 + 3\,600) \cdot 0{,}22 = 7\,392 \text{ (грн.)}$$

4. Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час виконання даного етапу роботи розраховуємо за формулою:

$$А = \frac{Ц \cdot Т}{12 \cdot Т_В} \quad (4.3)$$

де Ц – загальна балансова вартість обладнання, приміщення тощо, грн.;

Т – фактична тривалість використання, міс;

$Т_В$ – термін використання обладнання, приміщень тощо, роки.

Розробка програмного забезпечення ведеться 1,5 місяці.

Розрахунки зведено до таблиці 4.6:

Таблиця 4.6 – Амортизаційні відрахування

Найменування	Балансова вартість (грн.)	Термін використання (років)	Фактична тривалість використання, (міс.)	Величина амортизацій - них відрахувань, (грн..)
Офісне приміщення	50 000	20	1,5	625
Комп'ютер	10 000	5	1,5	500
Монітор	4 000	5	1,5	200
			Всього	1 325

5. Витрати на комплектуючі К, що були використані під час виконання даного етапу роботи, розраховуються за формулою:

$$К = \sum_{1}^{n} Н_i \cdot Ц_i \cdot К_i \text{ (грн.)} \quad (4.4)$$

де $Н_i$ – кількість комплектуючих *i*-го виду, шт.;

$Ц_i$– ціна комплектуючих *i*-го виду, грн.;

$К_i$ – коефіцієнт транспортних витрат, $К_i$ = (1,1…1,15);

n – кількість видів комплектуючих.

Таблиця 4.7 – Витрати на комплектуючі

Найменування комплектувальних	Кількість	Ціна за штуку, грн.	Сума, грн.	Примітка
Клавіатура (тип 1)	1	200 грн.	200 грн.	
Клавіатура (тип 2)	1	350 грн.	350 грн.	
Всього:			$К_i = 1,2$	550 грн.

6. Витрати на силову електроенергію $В_е$ розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п \text{ (грн.)} \quad (4.5)$$

Де В – вартість 1 кВт-год.;

П – установлена потужність обладнання – 0,8 кВт;

Ф – фактична кількість годин роботи обладнання – 480 годин,

$К_п$ – коефіцієнт використання потужності.

$$В_е = 3,5 \cdot 0,8 \cdot 480 \cdot 0,14 = 188 \text{ (грн).}$$

7. Інші витрати $В_{ін}$ охоплюють:

– витрати на управління організацією;

– оплату службових відряджень;

– витрати на утримання, ремонт та експлуатацію основних засобів;

– витрати на опалення, освітлення, водопостачання, охорону праці тощо.

Інші витрати $В_{ін}$ можна прийняти як 100% від суми основної заробітної плати розробника:

$$В_{ін} = 30\,000 \cdot 1 = 30\,000 \text{ (грн)}$$

Послуги Інтернету – 250 грн., канцтовари – 200 грн. Загальна вартість становить:

$$250 + 200 = 450 \text{ (грн.)}$$

8. Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи – В.

$$В = 30\,000 + 3\,600 + 7\,392 + 1\,325 + 550 + 188 + 30\,000 + 450 = 73\,505 \text{ (грн.)}$$

9. Проведемо прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи. Прогнозування здійснюється за формулою:

$$ЗВ = \frac{В_{заг}}{\beta}, \text{грн.} \quad (4.6)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Так, якщо розробка знаходиться:

– на стадії науково-дослідних робіт, то $\beta \approx 0{,}1$;

– на стадії технічного проектування, то $\beta \approx 0{,}2$;

– на стадії розробки конструкторської документації, то $\beta \approx 0{,}3$;

– на стадії розробки технологій, то $\beta \approx 0{,}4$;

– на стадії розробки дослідного зразка, то $\beta \approx 0{,}5$;

– на стадії розробки промислового зразка, $\beta \approx 0{,}7$;

– на стадії впровадження, то $\beta \approx 0{,}9$.

$В_{заг}$ – загальна вартість всієї наукової роботи.

$$В = 73\ 505\ (грн.)$$

$$ЗВ = \frac{73\ 505}{0{,}7} = 105\ 007\ (грн.)$$

Отже, прогноз загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи складає $105\ 007$ (грн.)

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

У даному підрозділі проведемо кількісне прогнозування, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної наукової роботи.

В умовах ринку узагальнюючим позитивним результатом, що його отримує підприємство від впровадження результатів тієї чи іншої розробки, є збільшення чистого прибутку підприємства. Зростання чистого прибутку можна оцінити у теперішній вартості грошей.

Зростання чистого прибутку забезпечить інвестору надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Виконання даної наукової роботи та впровадження її результатів складає приблизно 1 рік. Позитивні результати від впровадження розробки очікуються вже в перші місяці після впровадження.

Проведемо детальне прогнозування позитивних результатів та кількісне їх оцінювання по роках.

Обчислимо збільшення чистого прибутку підприємства $\Delta П_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta П_i = \sum_{1}^{n} (\Delta П_я \cdot N + П_я \cdot \Delta N)_i \quad (4.7)$$

де $\Delta П_я$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$П_я$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

Припустимо, що внаслідок впровадження результатів наукової розробки чистий прибуток підприємства збільшиться на 70 грн., а кількість одиниць реалізованої послуги збільшиться:

– протягом першого року – на 400 од.,

– протягом другого року – ще на 750 од.,

– протягом третього року – ще на 900 од.

Орієнтовно: реалізація продукції до впровадження результатів наукової розробки складала 1 шт., а прибуток, що його отримувало підприємство на одиницю продукції до впровадження результатів наукової розробки – 60 грн.

Потрібно спрогнозувати збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого прибутку підприємства $\Delta П_1$ протягом першого року складе:

$$\Delta П_1 = 60 \cdot 1 + (60 + 70) \cdot 400 = 76\,000 \text{ (грн.)}$$

Обчислимо збільшення чистого прибутку підприємства $\Delta П_2$ протягом другого року:

$$\Delta П_2 = 60 \cdot 1 + (60 + 70) \cdot (400 + 750) = 218\,500 \text{ (грн.)}$$

Збільшення чистого прибутку підприємства $\Delta П_3$ протягом третього року становитиме:

$$\Delta П_3 = 60 \cdot 1 + (60 + 70) \cdot (400 + 750 + 900) = 389\,500 \text{ (грн.)}$$

Отже, розрахунки показують, що відповідно прогнозуванню комерційний ефект від впровадження розробки виражається у значному збільшенні чистого прибутку підприємства.

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Основними показниками, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає:

1-й крок. Розрахунок теперішньої вартості інвестицій PV, що вкладаються в наукову розробку.

Такою вартістю ми можемо вважати прогнозовану величину загальних витрат ЗВ на виконання та впровадження результатів НДДКР, тобто $ЗВ = PV = 83\,439 \text{ (грн.)}$

2-й крок. Розрахуємо очікуване збільшення прибутку$\Delta П_i$, що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку також було розраховане нами раніше та становить:

$$\Delta П_1 = 76\,000(\text{грн}), \Delta П_2 = 218\,500(\text{грн}), \Delta П_3 = 389\,500(\text{грн}).$$

3-й крок. Будуємо вісь часу, на якій відображаємо всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Рисунок 4.1 характеризує рух платежів (інвестицій та додаткових прибутків).

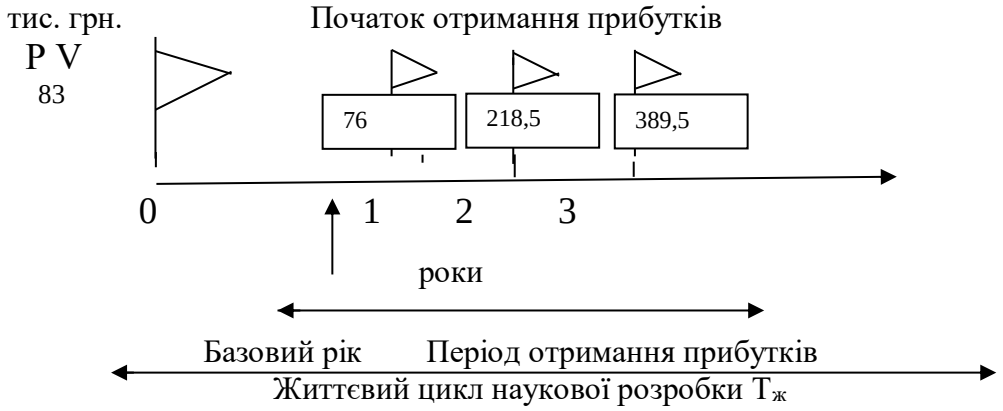


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

4-й крок. Розрахуємо абсолютну ефективність вкладених інвестицій $Е_{абс}$ за формулою:

$$Е_{абс} = (ПП - PV), (грн.) \quad (4.8)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій $PV = ЗВ$, грн.

Приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_{1}^{Т} \frac{\Delta П_i}{(1+\tau)^t}, (грн) \quad (4.9)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн.;

Т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні – 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки «0»;

$$\text{ПП} = \frac{76\,000}{(1+0{,}1)^1} + \frac{218\,500}{(1+0{,}1)^2} + \frac{389\,500}{(1+0{,}1)^3} = 542\,306{,}5\ (\text{грн.})$$

$$\text{Е}_{\text{абс}} = 542\,306{,}5 - 105\,007 = 437\,299{,}5\ (\text{грн.})$$

Оскільки $\text{Е}_{\text{абс}} > 0$, результат від проведення наукових досліджень щодо розробки програмного продукту та їх впровадження принесе прибуток, тобто є доцільним, проте це ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даної програми.

5-й крок. Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $\text{Е}_{\text{в}}$ за формулою:

$$\text{Е}_{\text{в}} = \sqrt[\text{Т}_{\text{ж}}]{1 + \frac{\text{Е}_{\text{абс}}}{PV} - 1} \qquad (4.10)$$

де $\text{Е}_{\text{абс}}$ – абсолютна ефективність вкладених інвестицій, грн.;

PV – теперішня вартість інвестицій $PV = \text{ЗВ}$, грн.

$\text{Т}_{\text{ж}}$ – життєвий цикл наукової розробки, роки.

$$E_{\text{в}} = \sqrt[3]{1 + \frac{437\,299{,}5}{105\,007}} - 1 = \sqrt[3]{5{,}2} - 1 = 0{,}71 \text{ або } 71\%$$

Порівняємо $E_{\text{в}}$ з мінімальною (бар'єрною) ставкою дисконтування τ_{min}, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть.

Спрогнозуємо величину τ_{min}.

У загальному вигляді мінімальна (бар'єрна) ставка дисконтування τ_{min}визначається за формулою:

$$\tau_{min} = d + f \tag{4.11}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; $d = 0{,}2$;

f – показник, що характеризує ризикованість вкладень; величина $f = 0{,}5$.

$$\tau_{min} = 0{,}2 + 0{,}5 = 0{,}7$$

Оскільки

$$E_{\text{в}} = 71\% > \tau_{min} = 70\%,$$

то у інвестора є потенційна зацікавленість у фінансуванні даної наукової розробки.

6-й крок. Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій $\text{Т}_{\text{ок}}$ за формулою:

$$\text{Т}_{\text{ок}} = \frac{1}{\text{Е}_{\text{в}}}, \text{рік} \tag{4.12}$$

$$\text{Т}_{\text{ок}} = \frac{1}{0{,}71} = 1{,}4 \text{ (року)}$$

Оскільки термін окупності вкладених у реалізацію наукового проекту інвестицій менше трьох років

$$(\text{Т}_{\text{ок}} < 3 \text{ років}),$$

то фінансування нової розробки є доцільним.

4.5 Висновки до розділу

В даному розділі було виконано оцінювання комерційного потенціалу розробки програмного засобу для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

Проведено технологічний аудит з залученням трьох незалежних експертів. Визначено, що рівень комерційного потенціалу розробки вище середнього. Проведено порівняння з аналогом. Згідно з проведеним оцінюванням нова розробка є якісною та конкурентоспроможною.

Рівень комерційного потенціалу розробки, становить 42,7 балів, що відповідає рівню «високий».

Згідно із розрахунками всіх статей витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи загальні витрати на розробку складають 105 007 (грн.). Розрахована абсолютна ефективність вкладених інвестицій в сумі 437 299,5 (грн.) свідчить про отримання прибутку інвестором від комерціалізації програмного продукту.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 71%, що вище за мінімальну бар'єрну ставку дисконтування, яка складає 70%. Це означає потенційну зацікавленість інвесторів у фінансуванні розробки.

Термін окупності вкладених у реалізацію проекту інвестицій становить 1,4 (року), що також свідчить про доцільність фінансування нової розробки.

Отже, проаналізувавши отримані економічні показники, можна вважати, що запропонована розробка програмного засобу має високий комерційний потенціал, а тому є доцільною для подальшого впровадження.

ВИСНОВОК

В магістерській дипломній роботі здійснювалась розробка програмного засобу для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

Вибір та поєднання даних стеганографічних методів зумовлені ефективністю результатів роботи: модифікованій метод вкраплення підвищує пропускну здатність контейнера, а схема розподілу покращує стійкість, як було показано в ході експериментальних досліджень.

В першому розділі було проведено теоретичний огляд галузі, в якій проводиться розробка. У даній роботі представлено основні поняття стеганографічних систем передачі інформації, також проведено аналіз класифікації стеганографічних методів та можливих атак на них.

В другому розділі здійснювалась робота над розробкою та проектуванням програмного додатку для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

В ході роботи було досліджено особливості алгоритм роботи методу найменш значущого біта та схеми розподілу секрету Шаміра, розроблено алгоритму роботи стегосистеми на основі обраних методів (алгоритм роботи програми для здійснення процесу вкраплення, алгоритм роботи програми для здійснення процесу вилучення даних).

Із врахуванням поставлених задач та особливостей розробки для її програмної реалізації було обґрунтовано вибір мови та середовища програмування.

В третьому розділі здійснювалась практична реалізація програмного додатку для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у

поєднанні із схемою розподілу секрету Шаміра.

В ході роботи було здійснено проектування користувацького інтерфейсу користувача, описано програмну реалізацію, детально описано інструкцію користувача для роботи з додатком.

Тестування практичних результатів показало, що модифікований алгоритм у поєднанні із схемою розподілу Шаміра більш коректно працює з міркувань пропускної здатності ніж звичайний НЗБ алгоритм, важче точно виявити обсяг вкрапленої інформації атаками та відновлення початкового повідомлення навіть після вилучення інформації у правильному порядку з контейнера.

В четвертому розділі роботи, проаналізувавши отримані економічні показники, можна вважати, що запропонована розробка програмного засобу має високий комерційний потенціал, а тому є доцільною для подальшого впровадження.

Таким чином, аналізуючи отримані результати практично реалізованої роботи,можна вважати, що в результаті дослідження було досягнути основну мету роботи, а саме розроблено та реалізовано програмний засіб для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

ПЕРЕЛІК ПОСИЛАНЬ

1. Комп`ютерна стеганографія: навчальний посібник / В.О. Хорошко, Ю.Є. Яремчук, В.В. Карпінець. Вінниця: ВНТУ, 2017. 155 с.

2. Основи комп’ютерної стеганографії : навч. посіб. / [В. О. Хорошко, О. Д. Азаров, М. Є. Шелест, Ю. Є. Яремчук]. — Вінниця : ВДТУ. — 2003. — 143 с.

3. Карпінець В. В. Вирішення проблеми погіршення якості векторних зображень при вбудовуванні цифрових водяних знаків / В. В. Карпінець, Ю. Є. Яремчук // Правове, нормативне, та метрологічне забезпечення системи захисту інформації в Україні — 2010. — № 1(20). — С. 72—82.

4. Карпінець В. В. Аналіз впливу цифрових водяних знаків на якість векторних зображень / В. В. Карпінець, Ю. Є. Яремчук // Сучасний захист інформації. — 2011. — № 1. — С. 72—82.

5. Карпінець В. В. Зменшення відхилень координат точок внаслідок вбудовування цифрових водяних знаків у векторні зображення / В. В. Карпінець, Ю. Є. Яремчук // Правове, нормативне, та метрологічне забезпечення СЗІ інформації в Україні — 2010. — № 2(21). — С. 101—109.

6. Jayaram P, Ranganatha H R, Anupama H S, " Information Hiding Using Audio Steganography – A Survey" in The International Journal of Multimedia & Its Applications (IJMA) Vol.3, No.3, August 2011

7. Jammi Ashok, Y. Raju, S. Munishankaraiah, K. Srinivas "Steganography: An Overview" in International Journal of Engineering Science and Technology Vol. 2(10), 2010

8. K.P.Adhiya and Swati A. Patil, " Hiding Text in Audio Using LSB Based Steganography" in Information and Knowledge Management Vol. 2, No.3, 2012

9. Pratap Chandra Mandal Modern "Steganographic technique: A survey" in International Journal of Computer Science & Engineering Technology (IJCSET)

10. Prof. Samir Kumar Bandyopadhyay and Barnali Gupta Banik "Multi-Level

Steganographic Algorithm for Audio Steganography using LSB Modification and Parity Encoding Technique" in International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) Volume 1, Issue 2, July – August 2012

11. R Sridevi, Dr. A Damodaram, DR. Svl. Narasimham "Efficient Method Of Audio Steganography By Modified LSB Algorithm And Strong Encryption Key With Enhanced Security" in Journal of Theoretical and Applied Information Technology

12. Vivek Jain , Lokesh Kumar, Madhur Mohan Sharma, Mohd Sadiq, Kshitiz Rastogi "Public-Key Steganography Based On Modified LSB Method" in Journal of Global Research in Computer Science Volume 3, No. 4, April 2012

13. S.Kaur, S.Bansal, R. K. Bansal, “Steganography and classification of image steganography techniques”, International Conference on Computing for Sustainable Global Development, India, 2014.

14. A. Abd EL-Latif, B.Abd-El-Atty, E. Venegas-Andraca, “A novel image steganography technique based on quantum substitution boxes”, Optics & Laser Technology, Volume 116, August 2019, Pages 92-102.

15. M.Hussain, A.Wahid Abdul Wahab, Y.Idris, S. Ho, K.Jung, “Image steganography in spatial domain: A survey”, Signal Processing: Image Communication, Volume 65, Pages 46-66, July 2018.

16. K.Gaurav, U.Ghanekar, “Image steganography based on Canny edge detection, dilation operator and hybrid coding”, Journal of Information Security and Applications, Volume 41, August 2018, Pages 41-51.

17. S.Kumar, A.Singh, M.Kumar, “Information hiding with adaptive steganography based on novel fuzzy edge identification”, Defense Technology, Volume 15, Issue 2, April 2019, Pages 162-169.

18. B.Feng, W.Lu, W.Sun, “Secure Binary Image Steganography Based on Minimizing the Distortion on the Texture”, IEEE Transactions on Information Forensics and Security, Volume 10, Issue 2, April 2015, Pages 243 - 255.

19. Кузнецов О. О. К89 Стеганографія: навчальний посібник / О.О.Кузнецов, С. П. Євсеєв, О. Г. Король. - Х.: Вид. ХНЕУ, 2011. - 232с. [Електронний ресурс]. Режим доступу: http://repository.hneu.edu.ua/jspui

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

21. Грибунин В.Г. Цифровая стеганография / В.Г.Грибунин, И.Н.Оков, И.В.Туринцев. – М. : СОЛОН-Пресс, 2002. – 272 с.

22. Быков С.Ф., Мотуз О.В. Основы стегоанализа.// Защита информации. Конфидент. – СПб.: 2000, № 3. – С. 38-41.

23. Грибунин В.Г., Оков И.Н., Туринцев И.В. Цифровая стеганография. – М.: Солон-Пресс, 2002. – 272 с.

24. Елтышева Е.Ю., Фионов А.Н. Построение стегосистемы на базе растровых изображений с учетом статистики младших бит // Вестник СибГУТИ. – 2009. № 1. – С. 67-84.

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

26. Жилкин М. Ю. Стегоанализ графических данных на основе методов сжатия // Вестник СибГУТИ. – 2008. № 2. – С. 62–66.

27. Генне О.В. Основные положения стеганографии.// Защита информации. №3, 2000.

28. N.F. Johnson, S. Jajodia, Steganalysis: The Investigation of Hidden Information, IEEE Information Technology Conference, Syracuse, New York, USA, Sept. 1st-3rd. 1998.

29. Барсуков В.С., Романцов А.П. Компьютерная стеганография: вчера, сегодня, завтра. Технологии информационной безопасности XXI века. [Електронний ресурс] Режим доступу: http://st.ess.ru/.

30. Пономарев К. И., Путилов Г. П. Стеганография: история и современные технологии. – М.: МИЭМ, 2009. – 70 с.

31. Алиев А. Т. Разработка моделей, методов и алгоритмов перспективных средств защиты информации в системах електронного документооборота на базе современных технологий скрытой связи: дис. … канд. техн. наук: 05.13.19. – Ростов-на-Дону: ЮФУ, 2008. – 216 с.

32. Мерзлякова Е. Ю. Построение стеганографических систем для

растровых изображений, базирующихся на теоретико-информационных принципах.:дис. канд. техн. наук: 05.13.19. – Новосибирск: СибГУТИ, 2011. – 161 с.

33. Жгун Т. В. Модель скрытой передачи информации в каналах связи: дис. … канд. ф.-мат. наук: 05.13.18. – В. Новгород: НовГУ, 2003. – 187 с.

34. Абазина Е. С. Метод скрытой передачи информации с кодовым уплотнением в видеоданных // Информация и космос. – 2014. – № 4. – С. 33–38.

35. Цветков К. Ю, Федосеев В. Е., Коровин В. М., Абазина Е. С. Модель кодера скрытых каналов с кодовым уплотнением с использованием сигнальных последовательностей Франка-Уолша, Франка-Крестенсона // Труды НИИР. – 2015. – № 1. – С. 2–11.

36. Аграновский А. В., Балакин А. В., Грибунин В. Г. , Сапожников С. А. Стеганография, цифровые водяные знаки и стегоанализ. Монография. – М.: Вузовская книга, 2009. – 220 с.

37. Небаева К. А. Разработка необнаруживаемых стегосистем для каналов с шумом: дис. … канд. тех. наук: 05.12.13. – СПб.: СПбГУТ, 2014. – 176 с.

38. Коржик В. И., Небаева К. А. Основы стеганографии: учебнометодическое пособие по выполнению практических занятий.– СПб.: СПбГУТ, 2015. – 20 с.

39. Цветков К. Ю, Федосеев В. Е., Абазина Е. С. Применение двумерных нелинейных сигналов Франка-Уолша, Франка-Крестенсона в методе формирования скрытых каналов с кодовым уплотнением в структуре сжимаемых видеоданных // Наукоемкие технологии в космических исследованиях Земли. – 2013. – №. 4. – С. 32–40.

40. Юдін О.К., Зюбіна Р.В., Фролов О.В. Аналіз стеганографічних методів приховування інформаційних потоків у контейнери різних форматів. - Pedram // 31st International Conference on Distributed Computing Systems Workshops. Minneapolis, USA, 2011. P. 1-6.

41. Lecture Notes on Cryptography / Goldwasser S., Bellare M. – Cambridge, Massachusetts, 2001. – 283 p.

42. Абазина Е. С., Ерунов А. А. Результаты моделирования метода скрытой передачи информации с кодовым уплотнением в видеоданных // Системы управления, связи и безопасности. – 2015. – № 2. – С. 1–25. [Електронний ресурс] Режим доступу: http://journals.intelgr.com/sccs/archive/

43. Цветков К. Ю. Методы цифровой стеганографии и их приложения в сетях спутниковой связи // Сборник трудов II военно-научной конференции Космических войск. – СПб.: МО РФ, 2004. – Т. 2. – С. 344–349.

44. Цветков К. Ю., Ефимов С. Н., Осташов И. Т. Защита инфокоммуникационных систем и сетей специального назначения: учебное пособие. – СПб.: ВКА имени А.Ф. Можайского, 2010. – 160 с.

45. Гонсалес Р., Вудс Р. Цифровая обработка изображений //М.: Техносфера. – 2012. – Т. 1104.

46. «Клієнт (інформатика)». [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Клієнт_(інформатика)

47. Darmstaedter V. Low Cost Spatial Watermaking / V. Darmstaedter, J. Delaigle, J. Quisquater // Computers and Graphics / V. Darmstaedter, J. Delaigle, J. Quisquater., 1998. – (Vol. 5) – C. 417-423.

48. Hsu C. DCT-based Watermaking for Video / C. Hsu, J. Wu. // IEEE Transactions on Consumer Electronics. – 1998. – С. 417–423.

49. Fridrich J. Combining Low-Frequency and Spread Spectrum Watermaking / J. Fridrich. // Proc. of the SPIE Conference on Mathematics of Data/Image Coding, Compression and Encryption.. – 1998. – №3 456. – С. 2–12.

50. Завьялов С. В. Стеганографические методы защиты информации: учебное пособие / С. В. Завьялов, Ю. В. Ветров. – Санкт-Петербург: Политехнический университет, 2012. – 190 с.

51. Рябко Б. Я. Основы современной криптографии и стеганографии / Б. Я. Рябко, А. Н. Фионов. – Москва: Горячая линия - Телеком, 2013. – 232 с. – (2-е издание).

52. Petitcolas F. Information Hiding: A Survey / F. Petitcolas, R. Anderson, M. Kuhn. // Proceedings of the IEEE. – 1999. – С. 1062–1078.

53. Коханович Г. Ф. Компьютерная стеганография. Теория и практика / Г. Ф. Коханович, А. Ю. Пузиренко. – Київ: МК-Пресс, 2006. – 288 с.

54. Steganos Software GmbH – Steganos Privacy Suite 11 [Електронний ресурс] / Steganos Software GmbH – 2014. – Режим доступу до ресурсу https://www.steganos.com/support/faq/index.php

55. CISSP – Steganography, An Introduction Using S-Tools [Електронний ресурс] // INFOSEC Institute. – 2015. – Режим доступу до ресурсу: http://resources.infosecinstitute.com/

56. Мясников А. ImageSpyer 2009 [Електронний ресурс] / А. Мясников // ProstoWeb. – 2009. – Режим доступу до ресурсу: http://www.prostoweb.com.ua/

57. Иванов В. Программа JSTEG [Електронний ресурс] / В. Иванов // Стеганография & путешествия. – 2013. – Режим доступу до ресурсу: http://www.nestego.ru/2013/01/jpeg-jsteg.html.

58. Kwan M. GIF colourmap steganography [Електронний ресурс] / M. Kwan // GifShuggle Home Page. – 2010. – [Електронний ресурс] Режим доступу до ресурсу: http://www.darkside.com.au/gifshuffle/.

59. Хьюбел Д. Глаз, мозг, зрение / Д. Хьюбел. – Санкт-Петербург: Слово, 1970. – 239 с.

60. Компьютерная анимация [Електронний ресурс] // Wikipedia. – 2014. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/

61. APNG [Електронний ресурс] // Wikipedia. – 2013. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/APNG.

62. Graphics Interchange Format [Електронний ресурс] Режим доступу: // Wikipedia. – 2015. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/GIF.

63. Краткое описание формата GIF [Електронний ресурс] Режим доступу: // Домашняя страничка Чьезо. – 2005. – Режим доступу до ресурсу: http://home.onego.ru/~chiezo/gif.htm.

64. Баричев С. Г. Стандарт AES. Алгоритм Rijdael / С. Г. Баричев, В. В. Гончаров, Р. Е. Серов // Основы современной криптографии / С. Г. Баричев, В. В. Гончаров, Р. Е. Серов. – Москва: Горячая линия - Телеком, 2002. – С. 30–35.

65. Шилдт Г. Java 8. Полное руководство / Г. Шилдт. – Москва: Вильямс, 2015. – 1397 с. – (9-е издание).

66. Скит Д. C# для профессионалов: тонкости программирования / Дж. Скит. – Москва: Вильямс, 2014. – 608 с. – (3-е издание).

67. Нейгел К. C# 5.0 и платформа .NET 4.5 для профессионалов / К. Нейгел, Б. Ивьен, Д. Глинн. – Москва: Диалектика, 2013. – 1440 с

68. Проектирование пользовательского интерфейса [Електронний ресурс] Режим доступу: https://ru.userinterface_page09

69. Метод RS [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/RS-анализ

70. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт / Уклад. В. О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.

ДОДАТКИ

Додаток А. Технічне завдання

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ
Голова секції «Управління інформаційною безпекою» кафедри МБІС
д.т.н., професор
_____ Ю. Є. Яремчук

«24»вересня 2021 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:

Підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра
08-42.МКР.008.00.00 ТЗ

Керівник магістерської кваліфікаційної роботи
д.т.н., проф. каф. МБІС Яремчук Ю.Є.
_____

Вінниця – 2021 р.

1 Найменування та область застосування

Програмний засіб для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

Область застосування: підвищення стійкості та пропускної здатності цифрових графічних контейнерів.

2 Підстави для розробки

Розробка виконується на основі наказу ректора ВНТУ 24 вересня 2021 року №277.

3 Мета та призначення розробки

3.1 Мета розробки

Метою роботи є розробка та реалізація програмного засобу для підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

3.2 Призначення

Розроблений програмний продукт забезпечує можливість підвищення стійкості та пропускної здатності цифрових графічних контейнерів вдосконаленням стеганографічного методу NZB-вкраплення у поєднанні із схемою розподілу секрету Шаміра.

4 Джерела розробки

1. Комп`ютерна стеганографія: навчальний посібник / В.О. Хорошко, Ю.Є. Яремчук, В.В. Карпінець. Вінниця: ВНТУ, 2017. 155 с.

2. Основи комп’ютерної стеганографії : навч. посіб. / [В. О. Хорошко, О. Д. Азаров, М. Є. Шелест, Ю. Є. Яремчук]. — Вінниця: ВДТУ. — 2003. — 143 с.

3. Карпінець В. В. Аналіз впливу цифрових водяних знаків на якість векторних зображень / В. В. Карпінець, Ю. Є. Яремчук // Сучасний захист інформації. — 2011. — № 1. — С. 72—82.

5 Вимоги до програми

5.1 Вимоги до функціональних характеристик

5.1.1 Програмний додаток призначений для підвищення стійкості та пропускної здатності цифрових графічних контейнерів.

5.1.2 Реалізація методу не повинна вимагати спеціальних ліцензійних програмних додатків

5.1.3 Програмний додаток повинен мати зручний, легкий у розумінні користувача інтерфейс.

5.2 Вимоги до надійності:

5.2.1 Програмний додаток повинен бути працездатним продуктом, функціонувати без помилок.

5.2.2 Програмний додаток повинен працювати без помилок, у випадку виникнення критичних ситуацій необхідно передбачити виведення відповідних повідомлень.

5.4 Вимоги до складу і параметрів технічних засобів:

– оперативна пам'ять – не менше 512 Mb.

5.5 Вимоги до інформаційної та програмної сумісності – будь-яка операційна система.

6 Вимоги до програмної документації

6.1 Обов’язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.3

7 Вимоги до технічного захисту інформації

6.1 Необхідно забезпечити підвищення стійкості та пропускної здатності цифрових графічних контейнерів.

8 Техніко-економічні показники

7.1 Програмний додаток має бути простим у використанні, легко змінюваним, мати можливість швидкого введення змін.

7.2 Витрати на програмні продукти, що використовуються в ході розробки мають бути мінімальними.

9 Стадії та етапи розробки

№	Назва етапів МКР	Початок	Закінчення
1	Визначення напрямку магістерської роботи, формулювання та затвердження теми	01.09.2021	26.09.2021
2	Аналіз предметної області обраної теми	27.09.2021	05.10.2021
3	Апробація отриманих результатів	06.10.2021	15.10.2021
4	Розробка алгоритму роботи	16.10.2021	31.10.2021
5	Написання магістерської роботи на основі розробленої теми	01.11.2021	14.11.2021
6	Розробка економічної частини	15.11.2020	21.11.2021
7	Передзахист магістерської кваліфікаційної роботи	22.11.2021	25.11.2021
8	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	26.11.2021	19.12.2021
9	Захист магістерської кваліфікаційної роботи	20.12.2021	23.12.2021

10 Порядок контролю та прийому

10.1 До приймання магістерської кваліфікаційної роботи надається:

– ПЗ до магістерської кваліфікаційної роботи;

– демонстрація результату магістерської кваліфікаційної роботи;

– презентація;

– відгук керівника роботи;

– відгук опонента.

Технічне завдання до виконання прийняв _____ Софина А.А.

Додаток Б. Лістинг (завантаження файлу – контейнера, задання секрету)

```
using Moserware.Security.Cryptography;
using System;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Windows.Forms;

namespace SecretSplitterWinForms
{
  public partial class Form1 : Form
  {
    public const string DialogFilter = "Split Secret Files (*.splitsecret)|*.splitsecret|GNU
Privacy Guard Files (*.gpg)|*.gpg|All Files (*.*)|*.*";
    public const string AllFilesFilter = "Text files (*.txt)|*.txt";
    private bool _HasAgreedToSafetyWarning;
    private Control _LastSelectedShareTypeControl;


    public enum Mode
    {
      Image,
      Audio
    }

    Image image;
    byte[] audio;
    byte[] file;
    string filename;
    Mode currentMode;
    Stopwatch stopwatch;

    public Form1()
    {
      InitializeComponent();
      Text                    +=                    "                    "                    +
Moserware.Security.Cryptography.Versioning.VersionInfo.CurrentVersion;
      //OutputConsole.Bind(console);
      // random.Checked = true;
      currentMode = Mode.Image;
      stopwatch = new Stopwatch();
    }
```

```
        private void button1_Click(object sender, EventArgs e)
        {
            if (image == null )
            {
                MessageBox.Show("Для вбудовування необхідно обрати зображення");
                return;
            }

            if (rdoHaveSecretMessage.Checked)
            {
                var shares = CreateSecretMessageShares();
                if (!String.IsNullOrEmpty(shares))
                {
                    txtShares.Text = shares;
                }
            }
            else if (rdoHaveSecretFile.Checked)
            {
                CreateSecretFileShares();
            }

        }
        private void ChangedSecretType(object sender, System.EventArgs e)
        {
            UpdateSecretPanels();
        }

        private void UpdateSecretPanels()
        {
            // I should be using databinding and other fancy things.. oh well :)
            var selectedShareType = rdoHaveSecretFile.Checked ? rdoHaveSecretFile :
rdoHaveSecretMessage;
            //btnCreateShares.Text = rdoHaveSecretFile.Checked ? "&Save Encrypted File and
Create Secret Pieces" : "&Split Message Into Pieces";
            if ((_LastSelectedShareTypeControl != selectedShareType) && (selectedShareType
== rdoHaveSecretFile))
            {
                GenerateRandomFileKey();
            }

            _LastSelectedShareTypeControl = selectedShareType;
            //pnlCreateMessage.Visible = rdoHaveSecretMessage.Checked;
            //pnlCreateSecretFileAdvancedInfo.Visible = rdoHaveSecretFile.Checked &&
chkShowAdvancedFileOptions.Checked;
            //pnlCreateSecretFileBasicInfo.Visible = rdoHaveSecretFile.Checked;

            // Set minimum window size
            // TODO: Should be smarter about this..
            //          int minHeight = 80 + tabCreate.Controls.Cast<Control>().Where(c =>
```

```
c.Visible).Select(c => c.Height).Sum();
                //    MinimumSize = new Size(MinimumSize.Width, minHeight);
                //Size = MinimumSize;
            }

            private void GenerateRandomFileKey()
            {
                // BONUS: you can specify however many characters you want ;)
                var match = Regex.Match(cboKeySizes.Text, @"(?<charCount>[0-
9]+)\s*characters\s*\((?<bitCount>[0-9]+)\s*bits\s*\)");
                if (!match.Success)
                {
                    ShowError("Specify a valid key size", "Invalid Key Size");
                    return;
                }

                var charSize = Int32.Parse(match.Groups["charCount"].Value);
                var bitSize = Int32.Parse(match.Groups["bitCount"].Value);

                // every char is 4 bits
                var charBitSize = charSize * 4;

                // If there is disagreement, pick the bigger one
                var maxBitSize = Math.Max(charBitSize, bitSize);

                txtMasterKey.Text                                                    =
HexadecimalPasswordGenerator.GeneratePasswordOfBitSize(maxBitSize);
            }
```

Додаток В. Лістинг (вбудовування секрету)

```
        private void CreateSecretFileShares()
            {
                if (!TryVerifyThresholdValues())
                {
                    return;
                }

                if (!File.Exists(txtSecretFilePath.Text))
                {
                    ShowError("Secret file does not exist", "File Not Found");
                    return;
                }

                if (String.IsNullOrEmpty(txtMasterKey.Text))
                {
                    ShowError("Для шифрування файлу потрібен дійсний ключ.", "Invalid Key
Length");
                    return;
                }

                byte[] keyBytes;
                if (!SecretEncoder.TryParseHexString(txtMasterKey.Text, out keyBytes))
                {
                    ShowError("The key must contain all hexadecimal characters.", "Invalid Key
Character");
                    return;
                }

                SplitSecret splitSecret = SecretSplitter.SplitFile(keyBytes, (int)nudThreshold.Value);

                saveFileDialog.Filter = DialogFilter;
                saveFileDialog.Title = "Save Encrypted File";
                saveFileDialog.InitialDirectory = Path.GetDirectoryName(txtSecretFilePath.Text);
                saveFileDialog.FileName                                                    =
Path.GetFileName(Path.ChangeExtension(txtSecretFilePath.Text, ".splitsecret"));
                saveFileDialog.AddExtension = true;

                ShowInfo("You'll now have to specify where to save the encrypted file.", "Specify
Encrypted File Path");
                if (saveFileDialog.ShowDialog() != DialogResult.OK)
                {
                    return;
                }

                splitSecret.EncryptFile(txtSecretFilePath.Text, saveFileDialog.FileName);
```

```
            var sb = new StringBuilder();
            sb.AppendLine("Here are your secret pieces that form the decryption key for the encrypted file located at:");
            sb.AppendLine(saveFileDialog.FileName);
            sb.AppendLine();
            int shareWidth = saveFileDialog.FileName.Length;

            foreach (var currentShare in splitSecret.GetShares((int)nudShares.Value))
            {
              string currentShareText = currentShare.ToString();
              sb.AppendLine(currentShareText);
              shareWidth = Math.Max(shareWidth, currentShareText.Length);
              sb.AppendLine();
            }

            SetWidthFromShareLength(shareWidth);

            sb.AppendLine("Розповсюджуючи секретні фрагменти, не забудьте також переконатися, що ви розповсюджуєте зашифрований файл. Зашифрований файл безпечно надіслати електронною поштою, але ви повинні безпечно розповсюдити секретні фрагменти");
            sb.AppendLine();
            sb.AppendLine("Переконайтеся, що кожна людина знає це точно " + ((int)nudThreshold.Value) + " фрагменти необхідні для відновлення файлу.");

            txtShares.Text = sb.ToString();
            //  tabs.SelectedTab = tabRecover;
          }

          private void SetWidthFromShareLength(int shareWidth)
          {
            var textSize = TextRenderer.MeasureText(new string('0', shareWidth), txtShares.Font);
            // HACK to get shares to to not wordwrap
            var calculatedWidthFromText = textSize.Width + 70;
            var screenBounds = Screen.FromControl(txtShares).Bounds;
            var screenLeft = Left - screenBounds.Left;
            var screenWidth = screenBounds.Width;
            Width = Math.Max(Math.Max(Math.Min(calculatedWidthFromText, screenWidth - screenLeft), MinimumSize.Width), Width);
          }

          private void ShowInfo(string text, string caption = null)
          {
            ShowMessage(MessageBoxIcon.Information, text, caption);
          }

          private void ShowError(string text, string caption = null)
          {
```

```
            ShowMessage(MessageBoxIcon.Error, text, caption);
        }

        private void ShowMessage(MessageBoxIcon icon, string text, string caption = null)
        {
            MessageBox.Show(this, text, caption, MessageBoxButtons.OK, icon);
        }

        private string CreateSecretMessageShares()
        {
            if (!TryVerifyThresholdValues())
            {
                return null;
            }

            if (String.IsNullOrWhiteSpace(txtSecretMessage.Text))
            {
                ShowError("Enter something for the secret message", "Empty secret");
                return null;
            }

            var sb = new StringBuilder();
            sb.AppendFormat("Your secret message has been split into the following {0}
pieces:", nudShares.Value);
            sb.AppendLine();
            sb.AppendLine();
            int maxShareWidth = 0;

            int index = 1;
            var extension = textBox1.Text.Split('.')[1];

            var extensionFile = "." + extension;

            saveFileDialog.FileName = index + extensionFile;

            DialogResult res = saveFileDialog.ShowDialog();

            if (res == System.Windows.Forms.DialogResult.OK)
            {

                foreach (var currentSplit in SecretSplitter.SplitMessage(txtSecretMessage.Text,
                    (int)nudThreshold.Value, (int)nudShares.Value))
                {
                    sb.AppendLine(currentSplit);
                    maxShareWidth = Math.Max(maxShareWidth, currentSplit.Length);
                    sb.AppendLine();

                    Embed(currentSplit, index, extensionFile);
```

```
                index++;
            }
        }

        MessageBox.Show("Вбудовування успішно завершено");

        SetWidthFromShareLength(maxShareWidth);

        sb.AppendFormat("To reconstruct your secret, you'll need to provide exactly {0} of the above pieces. Please remember to keep the pieces safe and give them only to people you trust.", nudThreshold.Value);
        sb.AppendLine();

        // tabs.SelectedTab = tabRecover;
        return sb.ToString();
    }

    private void Embed(string text, int index, string extensionFile)
    {
        Bitmap encrypted = Steganography.Steganography.InsertEncryptedTextToImage(pictureBox1.Image, text);

        if (encrypted != null)
        {
            var path = Path.GetDirectoryName(saveFileDialog.FileName);
            var dir = path + "\\result";

            if (!(Directory.Exists(dir)))
            {
                Directory.CreateDirectory(dir);
            }

            encrypted.Save($"{dir}\\output{index}{extensionFile}");

        }

    }
```

Додаток Г. Лістинг (вилучення секрету)

```
private bool TryVerifyThresholdValues()
    {
      if (nudThreshold.Value <= nudShares.Value)
      {
        return true;
      }

      ShowError("The number of required pieces must equal or exceed the number of pieces.", "Invalid minimum piece count");
      return false;
    }

    private void SetRecoverTextToExample()
    {
      inputText.Text = @"Майте на увазі, що ви повинні ввести лише мінімальну кількість секретних частин, необхідних для відновлення секрету.

Наприклад, лише 2 із наведених нижче секретних фрагментів необхідні для відновлення зразка секретного повідомлення:

[SAMPLE SECRET]
Видаліть один з наведених вище рядків і натисніть ""Recover Secret""";
      inputText.Text = inputText.Text.Replace("[SAMPLE SECRET]", CreateSecretMessageShares());
    }

    private void button2_Click(object sender, EventArgs e)
    {

    }

    private void rdoHaveSecretFile_CheckedChanged(object sender, EventArgs e)
    {
      UpdateSecretPanels();
    }

    private void rdoHaveSecretMessage_CheckedChanged(object sender, EventArgs e)
    {
      UpdateSecretPanels();
    }

    private void cboKeySizes_SelectedIndexChanged(object sender, EventArgs e)
    {

    }
```

```
        private void btnRecover_Click_1(object sender, EventArgs e)
        {
            if (String.IsNullOrWhiteSpace(inputText.Text))
            {
                if (MessageBox.Show(this, "No secret pieces were specified. Would you like to see an example?", "Example?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
                {
                    SetRecoverTextToExample();
                }
                return;
            }
            CombinedSecret combinedSecret;

            try
            {
                combinedSecret = SecretCombiner.Combine(inputText.Text);
            }
            catch (InvalidChecksumShareException invalidChecksumShareException)
            {
                // We can provide a little UI magic
                var badShareText = invalidChecksumShareException.InvalidShare;
                int ixBadShareStart = txtShares.Text.IndexOf(badShareText);
                inputText.SelectionStart = ixBadShareStart;
                inputText.SelectionLength = badShareText.Length;
                ShowError("The selected secret piece seems to have been typed incorrectly.");
                return;
            }
            catch (Exception exception)
            {
                if (MessageBox.Show(this, exception.Message + Environment.NewLine + Environment.NewLine + "Would you like to see an example? (If yes, the current textbox content will be replaced)", "Error", MessageBoxButtons.YesNo, MessageBoxIcon.Error) == DialogResult.Yes)
                {
                    SetRecoverTextToExample();
                }

                return;
            }

            if (combinedSecret.ShareType == SecretShareType.File)
            {
                openFileDialog.Filter = DialogFilter;
                openFileDialog.Title = "Open Encrypted File";
                openFileDialog.FileName = "";

                if (!_HasAgreedToSafetyWarning)
                {
```

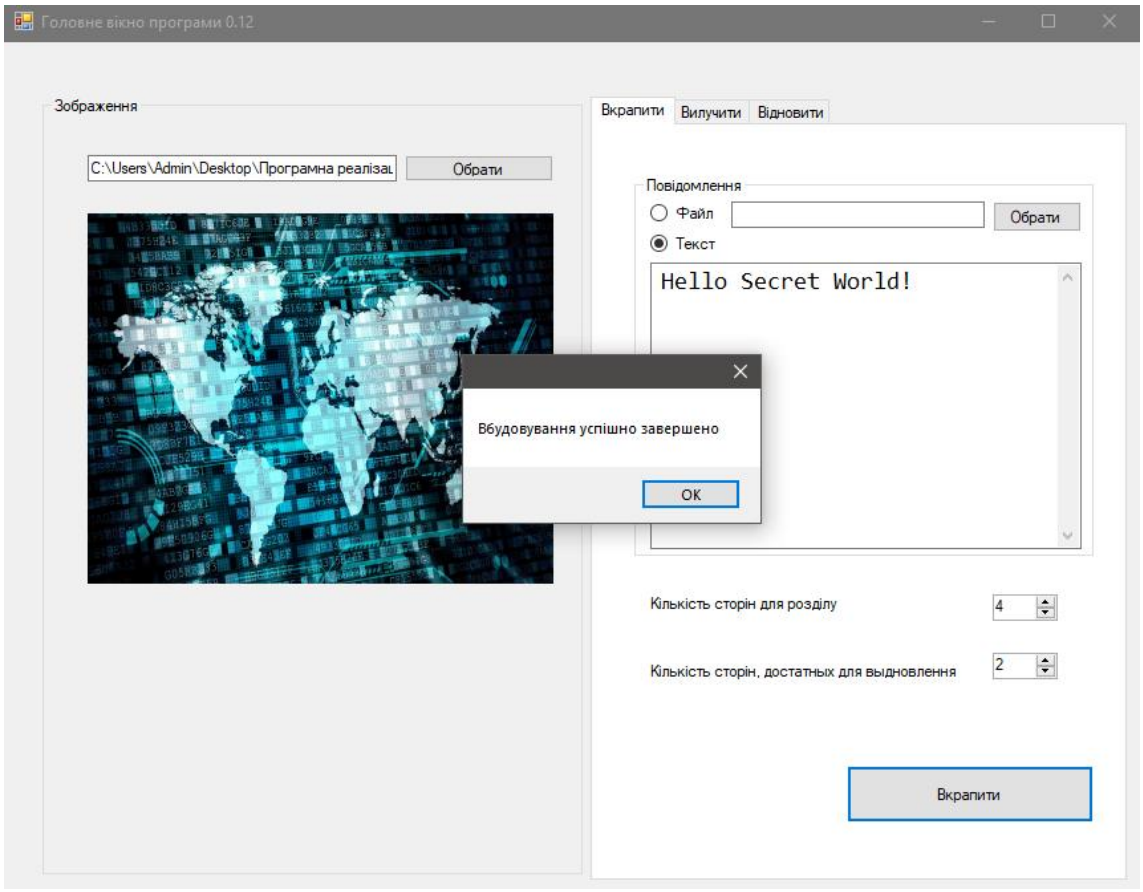
```
                ShowInfo("Схоже, ви намагаєтеся відновити файл. Тепер вам потрібно буде вказати зашифрований файл, щоб його розшифрувати. Цей зашифрований файл мав надати вам той, хто дав вам секретні фрагменти.», «Вкажіть файл для відкриття»");
            }

            if (openFileDialog.ShowDialog(this) != DialogResult.OK)
            {
              return;
            }

            try
            {
              using (var encryptedFileStream = File.OpenRead(openFileDialog.FileName))
              {
                string originalFileName;
                DateTime originalFileDate;
                var decryptedStream = combinedSecret.Decrypt(encryptedFileStream, out originalFileName,
                                                             out originalFileDate);

                if (!_HasAgreedToSafetyWarning)
                {
                  if (
                    MessageBox.Show(this,
                                    "Тепер вам потрібно буде вказати, де зберегти розшифрований файл. Розшифровані секретні файли зазвичай містять конфіденційну інформацію. Чи збережете ви розшифрований файл у безпечному місці (тобто на зашифрованому жорсткому диску або USB-накопичувачі, який ви знищите або безпечно видалите)?",
                                    "Чи збережете ви розшифровані дані в безпеці?", MessageBoxButtons.YesNo,
                                    MessageBoxIcon.Question) != DialogResult.Yes)
                  {
                    ShowError("Decryption aborted to protect privacy of contents.", "Unable to Decrypt");
                    return;
                  }
                  _HasAgreedToSafetyWarning = true;
                }

                saveFileDialog.DefaultExt = Path.GetExtension(originalFileName);
                saveFileDialog.Filter = String.Format("{0} files (*.{1})|*.{1}", saveFileDialog.DefaultExt.ToUpperInvariant(), saveFileDialog.DefaultExt) + "|" + AllFilesFilter;
                saveFileDialog.Title = "Save Decrypted File";

                saveFileDialog.FileName = Path.GetFileName(originalFileName);

                if (saveFileDialog.ShowDialog(this) != DialogResult.OK)
                {
```

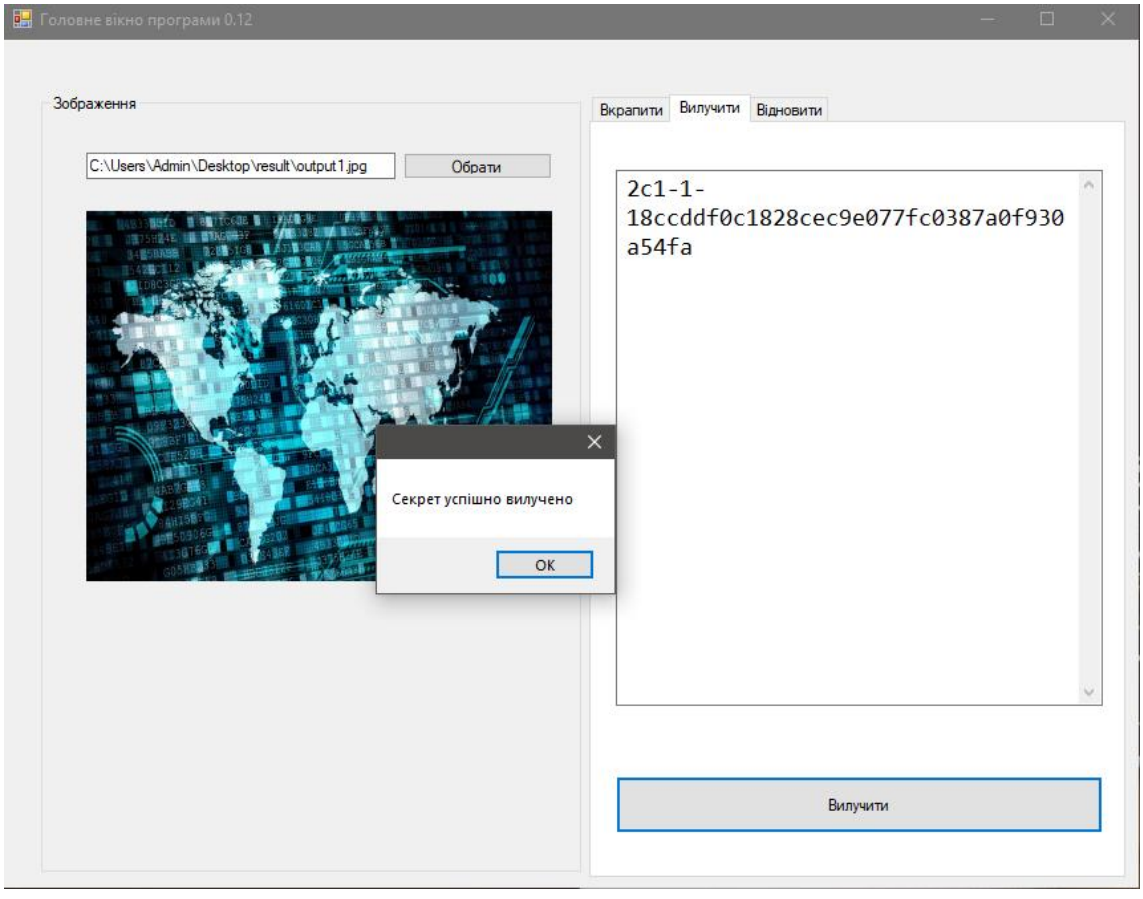
```
                    return;
                }

                if (File.Exists(saveFileDialog.FileName))
                {
                    File.Delete(saveFileDialog.FileName);
                }

                using (var decryptedFileStream = File.OpenWrite(saveFileDialog.FileName))
                {
                    decryptedStream.CopyTo(decryptedFileStream);
                }

                File.SetLastWriteTimeUtc(saveFileDialog.FileName,
originalFileDate.ToUniversalTime());

                if (MessageBox.Show(this, "Would you like to open the decrypted file?",
"Open Decrypted File?", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.Yes)
                {
                    Process.Start(new ProcessStartInfo(saveFileDialog.FileName) {
UseShellExecute = true });
                }
              }
            }
            catch (ModificationDetectedException modificationDetectedException)
            {
              ShowError("It looks like the file was tampered with or the given secret pieces
were invalid.",
                  "Modification Detected");
            }
            catch (Exception exception)
            {
              ShowError("Під час спроби розшифрувати файл сталася помилка. Будь
ласка, переконайтеся, що ви ввели лише найменшу кількість необхідних секретних частин,
вибрали правильний файл для розшифрування та чи файл не був підроблений.");
            }
          }
          else
          {
            rdoHaveSecretMessage.Checked = true;
            // tabs.SelectedTab = tabCreate;
            var recoveredTextString = combinedSecret.RecoveredTextString;

            if (!recoveredTextString.Any(c => Char.IsControl(c) && !Char.IsWhiteSpace(c)))
            {
              txtSecretMessage.Text = combinedSecret.RecoveredTextString;

              MessageBox.Show($"Відновлене повідомлення -
```

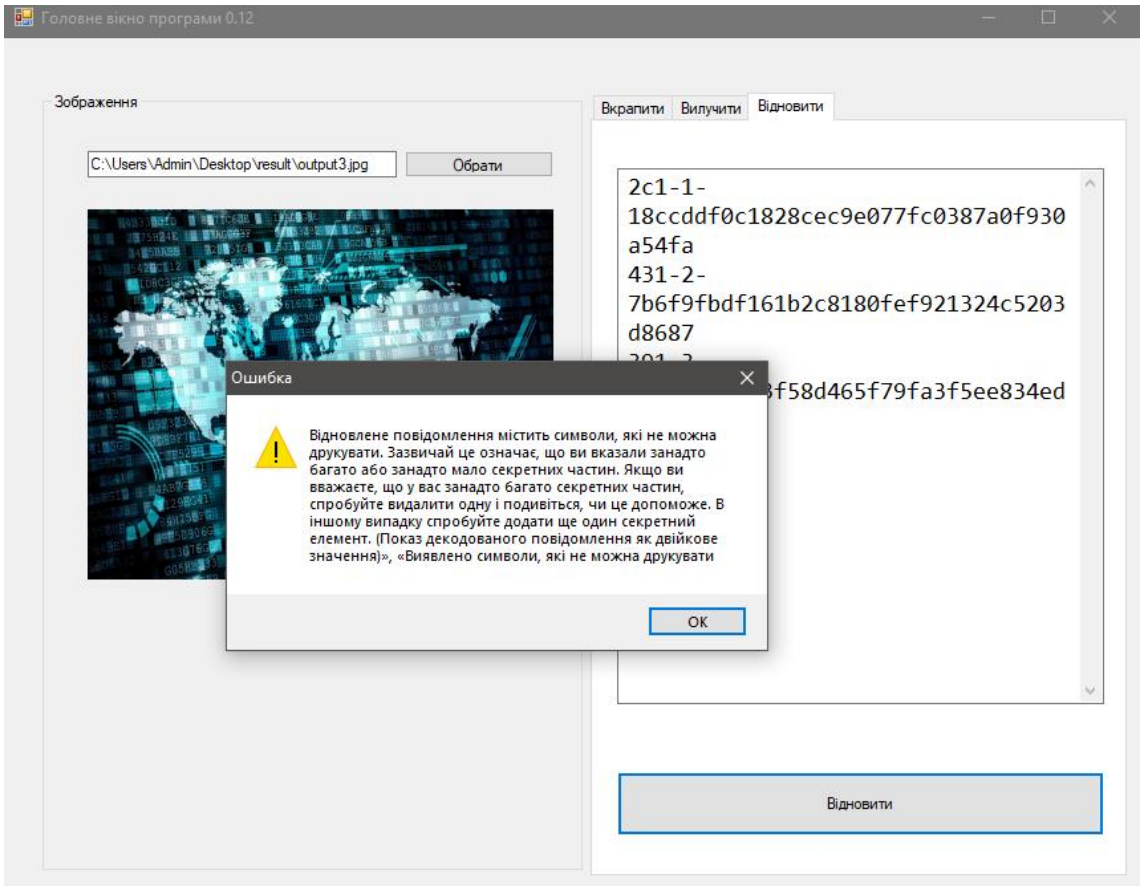
```
{combinedSecret.RecoveredTextString}");
                }
                else
                {
                    txtSecretMessage.Text = combinedSecret.RecoveredHexString;
                    ShowMessage(MessageBoxIcon.Warning, "Відновлене повідомлення містить символи, які не можна друкувати. Зазвичай це означає, що ви вказали занадто багато або занадто мало секретних частин. Якщо ви вважаєте, що у вас занадто багато секретних частин, спробуйте видалити одну і подивіться, чи це допоможе. В іншому випадку спробуйте додати ще один секретний елемент. (Показ декодованого повідомлення як двійкове значення)», «Виявлено символи, які не можна друкувати");
                }

                txtSecretMessage.SelectAll();
            }
        }

        private void Form1_Load_1(object sender, EventArgs e)
        {
            //  HACK to get reasonable height
            //  tabs.SelectedTab = tabCreate;
            UpdateSecretPanels();
            cboKeySizes.SelectedIndex = 0;
            //tabs.SelectedTab = tabRecover;
        }

        private void btnBrowsePlaintext_Click(object sender, EventArgs e)
        {
            openFileDialog.Filter = AllFilesFilter;
            openFileDialog.Title = "Виберіть файл, що містить секретну інформацію";
            openFileDialog.CheckFileExists = true;
            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                txtSecretFilePath.Text = openFileDialog.FileName;
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {

            openFileDialog.FileName = "*.*";
            DialogResult res = openFileDialog.ShowDialog();
            if (res == System.Windows.Forms.DialogResult.OK)
            {
                if (image != null)
                {
                    image.Dispose();
                }
                string ext = Path.GetExtension(openFileDialog.FileName);
```

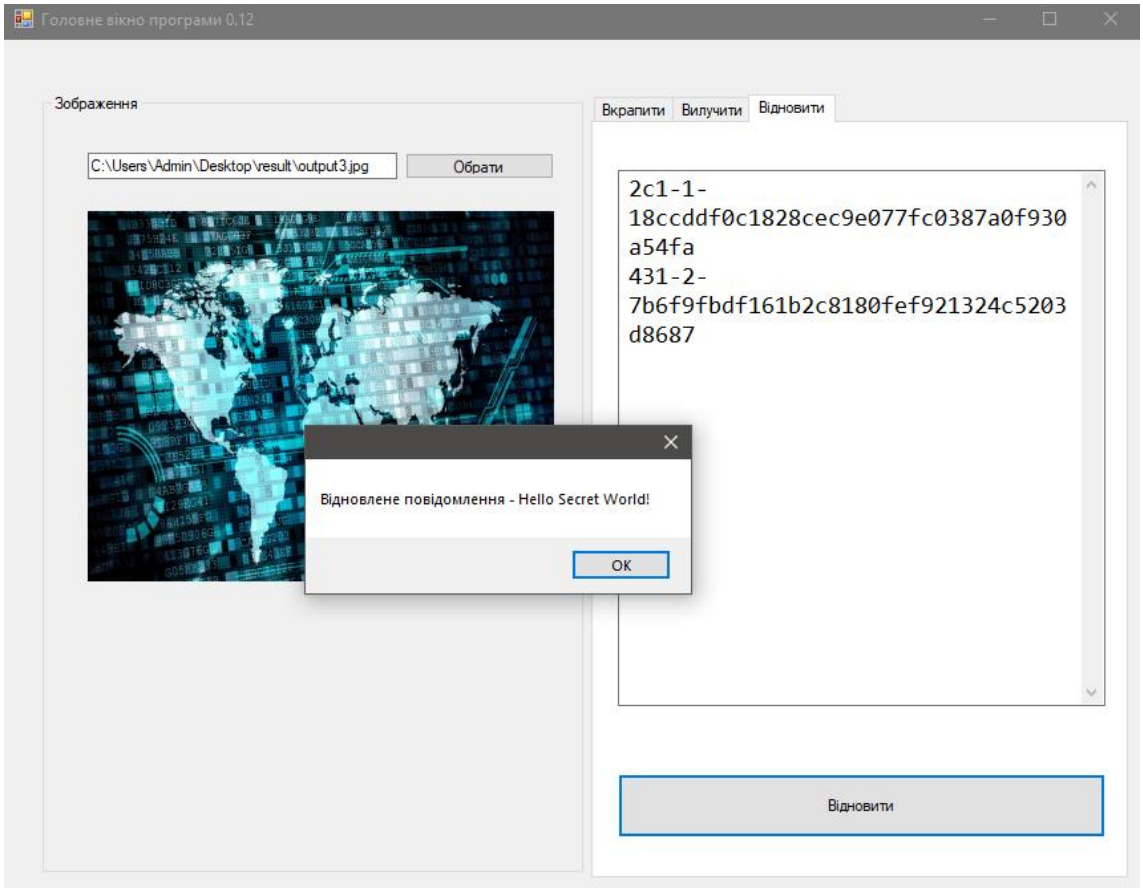
```
                if (ext == ".png" || ext == ".bmp" || ext == ".jpg")
                {
                    try
                    {
                        image = Image.FromFile(openFileDialog.FileName);
                        textBox1.Text = openFileDialog.FileName;
                        pictureBox1.Image = image;
                        currentMode = Mode.Image;
                        audio = null;
                        // audioLabel.Visible = false;
                    }
                    catch
                    {
                        image = null;
                        pictureBox1.Image = null;
                    }
                }
            }
        }

        private void decryptButton_Click_1(object sender, EventArgs e)
        {
            if (currentMode == Mode.Image && image != null)
            {
                string text = Steganography.Steganography.GetDecryptedTextFromImage(image);

                if (text != null)
                {
                    textBox3.Text = text;
                    MessageBox.Show("Секрет успішно вилучено");
                }
                else
                {
                    MessageBox.Show("Це зображення не містить секрету або сталася
помилка");
                }
            }

        }
    }
}
```

Додаток Д. Діалогові вікна додатку






Головне вікно програми 0.12
Зображення
C:\Users\Admin\Desktop\result\output3.jpg
Обрати
Вкрапити
Вилучити
Відновити
2c1-1-
18ccddf0c1828cec9e077fc0387a0f930
a54fa
431-2-
7b6f9fbdf161b2c8180fef921324c5203
d8687
f58d465f79fa3f5ee834ed
Ошибка
Відновлене повідомлення містить символи, які не можна друкувати. Зазвичай це означає, що ви вказали занадто багато або занадто мало секретних частин. Якщо ви вважаєте, що у вас занадто багато секретних частин, спробуйте видалити одну і подивіться, чи це допоможе. В іншому випадку спробуйте додати ще один секретний елемент. (Показ декодованого повідомлення як двійкове значення)», «Виявлено символи, які не можна друкувати
OK
Відновити


Головне вікно програми 0.12
Зображення
C:\Users\Admin\Desktop\result\output3.jpg
Обрати
Вкрапити
Вилучити
Відновити
2c1-1-
18ccddf0c1828cec9e077fc0387a0f930
a54fa
431-2-
7b6f9fbdf161b2c8180fef921324c5203
d8687
Відновлене повідомлення - Hello Secret World!
OK
Відновити

Додаток Е. Ілюстративний матеріал

Додаток Ж. Протокол перевірки