Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет «Комп'ютерні технології і системи»
(назва факультету)

Кафедра «Електронні обчислювальні машини»
(повна назва кафедри)

Пояснювальна записка

до кваліфікаційної роботи
бакалавра
(ступінь вищої освіти)

на тему: Розробка комплексу засобів стеганографічного захисту інформації. Стеганографічний захист інформації з використанням звукових файлів-контейнерів

за освітньою програмою Кібербезпека
зі спеціальності: 125 Кібербезпека
(шифр і назва спеціальності)

Виконав: студент групи: КБ1811

(підпис студента) / Денис ЗАЙЦЕВ /
(Ім'я ПРІЗВИЩЕ)

Керівник: (підпис) / доцент, Денис ОСТАПЕЦЬ
(посада, Ім'я ПРІЗВИЩЕ)

Нормоконтролер: (підпис) / ст. викладач, Володимир ДЗЮБА /
(посада, Ім'я ПРІЗВИЩЕ)

Консультанти:

(назва розділу) (підпис) / (посада, Ім'я ПРІЗВИЩЕ) /

(назва розділу) (підпис) / (посада, Ім'я ПРІЗВИЩЕ) /

(назва розділу) (підпис) / (посада, Ім'я ПРІЗВИЩЕ) /

Засвідчую, що у цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент (підпис)

Дніпро –2022 рік

Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies

Faculty «Computer technologies and systems»
(faculty)

Department «Electronic computers»
(department)

Explanatory Note
to Bachelor's Thesis
first (bachelor's)
(higher education degree)

on the topic: Development of a set of means of steganographic protection of information. Steganographic protection of information using audio file-containers

according to educational curriculum Cybersecurity

in the Speciality: 125 Cybersecurity
(speciality and its code)

Done by the student of the group: KB1811 / Denys Zaytsev /
(name, surname)

Scientific Supervisor: / Associate Professor, Denis Ostapets /
(position, name, surname)

Normative controller : / Senior lecturer, Volodymyr Dziuba /
(position, name, surname)

Supervisors

(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)
(Chapter title heading)	/ / (position, name, surname)

Dnipro – 2022

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: Комп'ютерні технології і системи
Кафедра: ЕОМ
Рівень вищої освіти: Перший (бакалаврський)
Освітня програма: Кібербезпека
Спеціальність: 125 Кібербезпека
(шифр та назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри _____
_____ _____
(підпис) (Ім'я ПРІЗВИЩЕ)
Дата _____

З А В Д А Н Н Я

на кваліфікаційну роботу бакалавра
(ступінь вищої освіти,

студенту Зайцеву Денису Дмитровичу
(Прізвище, Ім'я По батькові)

1. Тема роботи: Розробка комплексу засобів стеганографічного захисту інформації. Стеганографічний захист інформації з використанням звукових файлів-контейнерів

Керівник роботи: Остапець Денис Олександрович, к.т.н, доцент
(Прізвище, Ім'я, По батькові, науковий ступінь, вчене звання)

затверджені наказом від "07" 12 2021 р. № 67ст

2. Строк подання студентом роботи: 13.06.2022 р.

3. Вихідні дані до роботи: Методи стеганографічного захисту інформації; Формати звукових файлів-контейнерів

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):

4.1 Аналітична частина:

Аналіз методів стеганографії, що використовують звукові контейнери

4.2 Основна частина:

- Огляд методів та засобів стеганографічного захисту інформації;
- Режими роботи та інформаційна структура комплексу;
- Розробка програмного забезпечення

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):

- Порівняльний аналіз методів стеганографії;
- Склад та функції комплексу;
- Структура даних;
- Основні алгоритми програми;
- Приклади роботи комплексу

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Завдання видав: (підпис консультанта, дата)	Завдання прийняв (підпис студента, дата)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Приміт
1	Огляд методів та засобів стеганографічного захисту інформації	25.04.22	20%
2	Режими роботи та інформаційна структура комплексу	11.05.22	30%
3	Розробка та налагодження програмного забезпечення	06.06.22	45%
4	Реферат, вступ, висновки	13.06.22	5%
5	Подання кваліфікаційної роботи до кафедри	13.06.22	
6	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	23.06.22	

Студент _____ (підпис) Денис ЗАЙЦЕВ (Ім'я ПРІЗВИЩЕ)

Керівник роботи _____ (підпис) Денис ОСТАПЕЦЬ (Ім'я ПРІЗВИЩЕ)

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Завдання видав: (підпис консультанта, дата)	Завдання прийняв: (підпис студента, дата)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Огляд методів та засобів стеганографічного захисту інформації	25.04.22	20%
2	Режими роботи та інформаційна структура комплексу	11.05.22	30%
3	Розробка та налагодження програмного забезпечення	06.06.22	45%
4	Реферат, вступ, висновки	13.06.22	5%
5	Подання кваліфікаційної роботи до кафедри	13.06.22	
6	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	23.06.22	

Студент _____ Денис ЗАЙЦЕВ
(підпис) (Ім'я ПРІЗВИЩЕ)

Керівник роботи _____ Денис ОСТАПЕЦЬ
(підпис) (Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи бакалавра: 59 с., 22 рис., 3 табл., 9 додатків, 12 джерел.

Об’єкт розробки – засоби стеганографічного захисту інформації з використанням звукових файлів-контейнерів за методом найменшого значущого біта.

Мета роботи – створення програмного комплексу, що реалізує та демонструє стеганографічний захист інформації з використанням звукових файлів-контейнерів.

Здійснено огляд та аналіз засобів демонстрації стеганографічного захисту інформації з використанням звукових файлів-контейнерів. Обрано метод найменшого значущого біту та контейнер формату «wav», розглянуто його структуру. Наведені узагальнені алгоритми роботи комплексу в різних режимах, розроблено програмне забезпечення комплексу та перевірена його працездатність, написано інструкцію з використання.

Розроблене програмне забезпечення може використовуватися для прихованого обміну даними та у навчальному процесі.

Ключові слова: СТЕГАНОГРАФІЯ, ЗАХИСТ ІНФОРМАЦІЇ, АУДІО ФАЙЛИ, WAV, МЕТОД НАЙМЕНШОГО ЗНАЧУЩОГО БІТУ, LSB, C#, WPF, AES 128

ЗМІСТ

ВСТУП

Зараз ми живемо в інформаційному суспільстві, де безпечний обмін інформацією виходить на перший план. Кожен рік засоби зв'язку удосконалюються, а тому вимагають розробки спеціальних засобів безпечного зберігання та передачі інформації. Конфіденційність може забезпечити така наука, як криптографія, але є важливі задачі інформаційної безпеки, яку неможливо розв’язати тільки методами криптографії. Однією з таких задач є приховування самого факту існування конфіденційної інформації або її передачі. Саме ця задача вирішується за допомогою методів стеганографії. Під час розвитку технологій у сучасної стеганографії з’явилося відгалуження під назвою цифрова стеганографія. Це направлення основане на використанні цифрових об’єктів, таких як зображення, текстові файли, мережеві пакети та інші, які застосовуються в якості оболонки для секретної інформації.

Безпечний обмін конфіденційною інформацією у мережі, її надійне зберігання є важливим питанням у сучасному світі. Тому тема роботи є актуальною.

Мета роботи – створення програмного комплексу, що реалізує та демонструє стеганографічний захист інформації з використанням звукових файлів-контейнерів.

Основні положення роботи доповідалися та були схвалені на XV Міжнародній конференції «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» у 2021 році (див. додаток А).

1 ОГЛЯД МЕТОДІВ ТА ЗАСОБІВ СТЕГАНОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ

1.1 Загальні відомості

Стенографія — це метод приховування секретних даних у звичайному, несекретному файлі чи повідомленні, щоб уникнути їхнього виявлення. Це один із способів захисту інформації, де саме зберігання інформації або її передача залишається таємною [1].

Робота стосується сучасної стеганографії, а саме про її відгалуження цифрової стеганографії. Це направлення основане на використанні цифрових об’єктів, таких як зображення, текстові файли, мережеві пакети та інші, які застосовуються в якості оболонки для секретної інформації. Для огляду методів та засобів цифрової стеганографії розглянемо основні поняття для сучасних стеганографічних систем відповідно до [2]:

Стеганографічна система (стегосистема) – це поєднання методів та засобів, що використовуються для створення прихованих каналів для передачі інформації. Тобто це весь інструментарій, що потрібен для того, щоб була можливість обмінюватися прихованою інформацією.

Повідомлення – це інформація, яку один абонент бажає передати іншому абоненту. У нашому випадку ця інформація подана неявно.

Контейнер (стегоконтейнер) – вихідний файл, що використовується для приховування повідомлень. Стегоконтейнери окремо поділяються на пустий, який не містить секретних даних та на заповнений, в якому є прихована інформація.

Стеганографічний канал – канал, через який фактично передаються заповнені або пусті стеганографічні контейнери.

Обсяг стегоконтейнера – найбільша з можливих частин файлу, придатна для вбудовування повідомлення. Залежить від методу вбудовування.

1.2 Огляд методів стеганографії з використанням звукових контейнерів

Метод LSB полягає у використанні похибки дискретизації, яка завжди існує в оцифрованих зображеннях або аудіо- та відеофайлах [3]. Ця похибка дорівнює найменш значущій цифрі числа, що визначає розмір елемента файлу. Тому в більшості випадків модифікація найменш значущого біта не призведе до значного перетворення файлів. Схема роботи представлена на рис. 1.1.

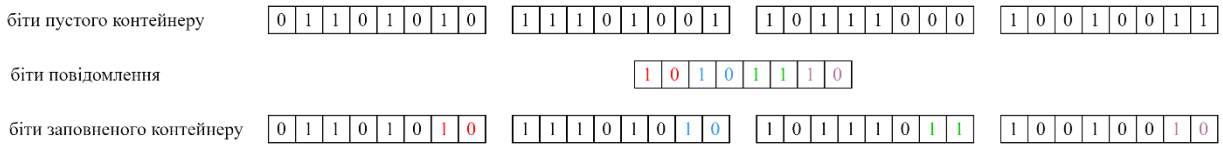

Рисунок 1.1 – Схема роботи методу LSB

Цей метод має декілька особливостей. Його найлегше впізнати, у міру зміни інформації статистичні характеристики цифрового потоку будуть спотворюватися а отже, ці ознаки потребують виправлення. До переваги можна віднести велику кількість інформації яку можна приховувати та передавати.

Метод фазового кодування передбачає використання слабкої слухової чутливості людини до незначних змін фази сигналу. Фаза початкового сегмента звукового сигналу буде змінена на приховане повідомлення [4]. Надалі фази сегментів мають бути узгоджені з сегментом у якому приховане повідомлення для підтримки різниці фаз. Цей метод є одним з найефективніших методів з точки зору відношення сигнал / шум, тобто з точки зору вбудованого сигналу та рівня перешкод, які ми відтворюємо.

Під час виконання методу фазового кодування звуковий сигнал розділяється на невеликі сегменти, використовується перетворення Фур'є для створення масивів фаз та амплітуд і розраховується різниця фаз між сусідніми сегментами. Після цього створюється новий масив фаз, звуковий сигнал відновлюється на основі зворотного дискретного перетворення Фур'є.

Ехо – метод дозволяє вбудовувати дані в сигнал, змінюючи параметри ехо сигналу. Параметри ехо, що несуть вбудовану інформацію, включають: початкову амплітуду, час спаду та зсув (затримка часу між вихідним сигналом та

його ехо). Коли зсув зменшується, два сигнали змішуються. У якийсь момент людське вухо перестає розрізняти ці два сигнали, і ехо сприймається як додатковий резонанс. Цей момент важко визначити, оскільки це залежить від оригінального запису, типу звуку та аудиторії.

Метод передбачає використання двох затримок, одна для кодування нуля, інша для кодування одиниці. Вихідний сигнал ділиться на невеликі сегменти. Кожен сегмент вважається окремим сигналом, в який вбудована частина повідомлення, тобто є невелика затримка часу. Потім ці сегменти об'єднуються для формування заповненого контейнеру [5].

Метод розширення спектру прямою послідовністю розширює сигнал повідомлення шляхом множення цього сигналу на псевдовипадкову послідовність максимальної довжини, що промодульована відомою частотою.

Метод розширення спектру прямою послідовністю для аудіофайлів полягає в наступному. Сигнал повідомлення помножується на сигнал з псевдовипадковою шумовою послідовністю, що характеризується широким спектром частот. В результаті спектр повідомлення розширюється, щоб охопити всю доступну пропускну здатність. Далі послідовність розширеного повідомлення послаблюється і додається до вихідного сигналу як додатковий випадковий шум [6].

1.3 Огляд деяких засобів стеганографії

Засіб **OpenPuff** включає приховану стеганографію, унікальні рівні безпеки та кілька форматів медіа, таких як зображення (BMP, JPG, PCX, PNG, TGA), аудіофайли (AIFF, MP3, NEXT / SUN, WAV), відеофайли (3GP, MP4, MPG, VOB) та Adobe – файли (FLV, SWF, PDF).

Засіб підтримує два основних сценарії роботи:

- Приховування в файл повідомлення з шифруванням;
- Підпис файлу.

Особливістю цього засобу є можливість розбиття повідомлення на окремі частини, якщо це повідомлення більше ніж обсяг одного стегоконтейнера (див. рис. 1.2). Створюється ланцюжок з декількох контейнерів та повідомлення розфасовується на весь цей ланцюжок. При цьому дістати правильні дані можливо тільки тоді, коли отримувач правильно вкаже послідовність з стегоконтейнерів.

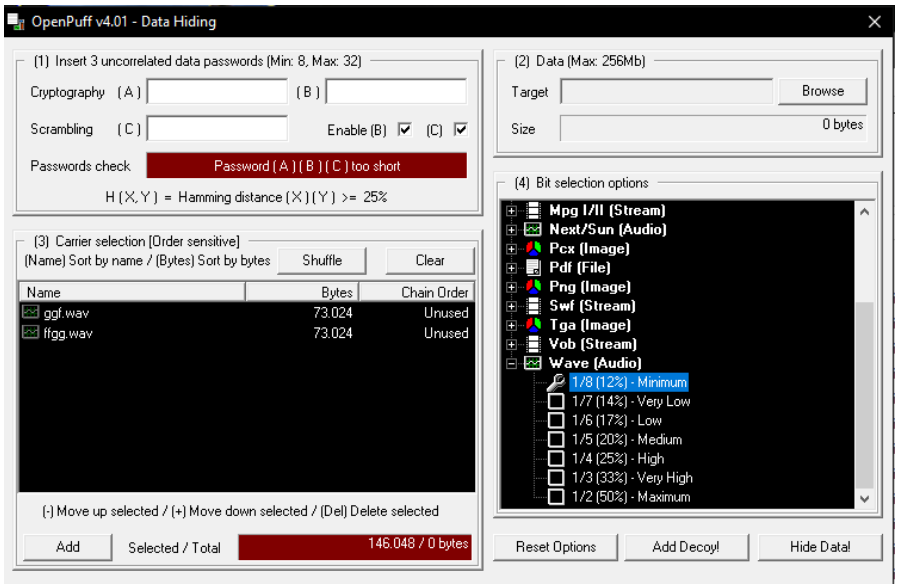

Рисунок 1.2 - Параметри для приховування повідомлення

При використанні програми була перевірена якість приховування інформації в WAV-файлах. Повідомлення, що вбудоване в сигнал не вносить спотворення, яке може почути людина. Також використовуючи цей засіб можливо задати обсяг стегоконтейнера.

Засіб **DeepSound** ховає повідомлення тільки всередині звукових файлів. DeepSound може використовувати WAV (без стиснення, лише PCM) як контейнер, а також MP3, CDA, WMA, APE та FLAC. DeepSound може вбудовувати файли будь-якого типу та автоматично обчислювати їх вільний простір на основі розміру контейнера та налаштувань якості звуку. Інтерфейс програми представлений на рис. 1.3.

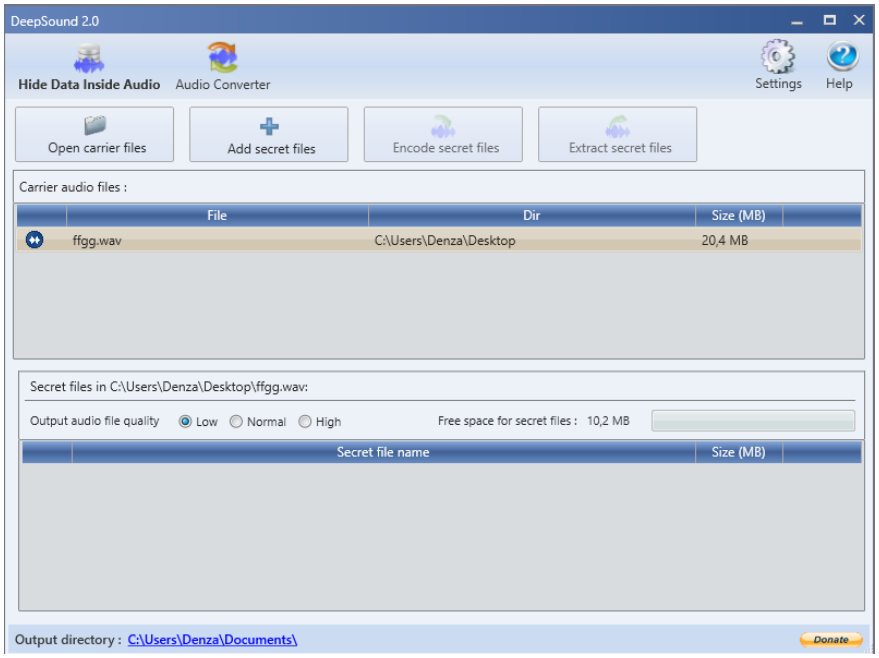

Рисунок 1.3 - Інтерфейс DeepSound

Цей засіб на відміну від OpenPuff дає можливість сховати набагато більше повідомлення, але за це платить якістю вихідного сигналу, тож людина здатна почути шуми вже з встановленою якістю звуку «Normal».

1.4 Порівняльний аналіз методів стеганографії з використанням звукових контейнерів

В таблиці 1.1 автором дається порівняння методів, що були розглянуті у пункті 1.2.

Таблиця 1.1 – Порівняння стеганографічних методів

Характеристики / Методи	Відносна помітність на слух	Відносна стійкість до стегоаналізу	Обсяг	Відносна залежність від формату файлу	Відносна простота реалізації
LSB	LOW	LOW	HIGH	HIGH	HIGH
Фазове кодування	LOW	MED	LOW	LOW	MED
Ехо-кодування	MED	MED	MED	LOW	MED
Розширення спектру прямою послідовністю	HIGH	HIGH	MED	MED	LOW

В методі фазового кодування на повідомлення змінюється лише фаза першого сегменту, через що обсяг стегоконтейнера є невеликим. Метод не залежить від формату аудіофайлу.

Ехо-кодування має високу помітність на слух, а тому вирішено його не використовувати.

Метод розширення спектру прямою послідовністю складний у реалізації порівнянно з іншими, але стійкий до стегоаналізу.

Метод LSB найпростіший у реалізації, має великий обсяг для приховування повідомлення порівняно з іншими методами, та відносно непомітність на слух. Тому серед розглянутих методів приховування інформації в аудіо файлах, для реалізації обрано саме метод найменшого значущого біта.

1.5 Висновки за розділом

У даному розділі наведені визначення основних понять сучасної стеганографії. Розглянуті та проаналізовані методи цифрової стеганографії для звукових контейнерів та результатом аналізу для реалізації в роботі обрано метод LSB. Також наведено огляд відомих програмних засобів аудіостеганографії, серед яких були OpenPuff та DeepSound.

2 РЕЖИМИ РОБОТИ ТА ІНФОРМАЦІЙНА СТРУКТУРА КОМПЛЕКСУ

2.1 Функціонування комплексу

У роботі вирішено реалізувати програму, яка повинна виконувати функції приховування таємного повідомлення у аудіоконтейнері та його вилучення. Роботу комплексу графічно представлено на рис. 2.1.

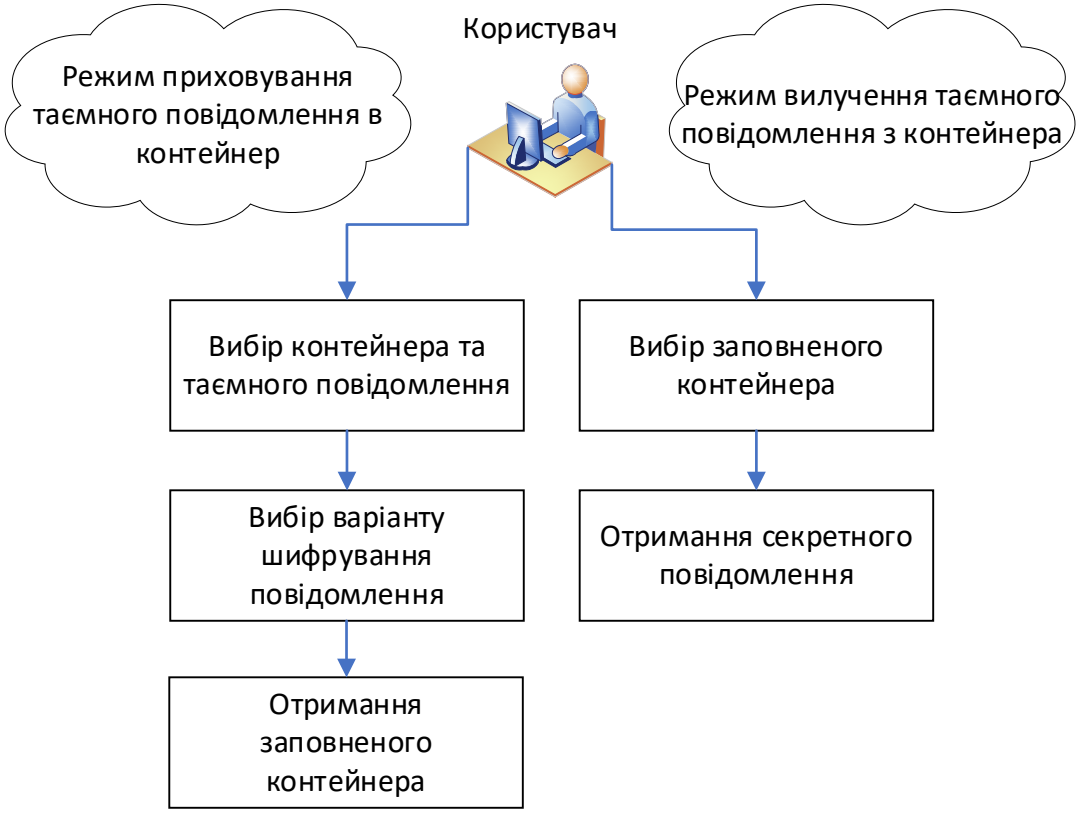

Рисунок 2.1 - Робота комплексу в режимах приховування та вилучення таємного повідомлення

Як було указано у розділі 1, даний комплекс повинен реалізувати метод найменшого значущого біта (LSB). Недоліком ж є його популярність, завдяки якій при виявленні стегоканалу зловмисником той зможе з легкістю виявити секретне повідомлення. Для вирішення цієї проблеми у програмі також буде реалізовано можливість шифрування секретного повідомлення за алгоритмом AES 128.

2.2 Структура файлу контейнеру

У якості контейнера обрано аудіо файли з розширенням «.wav». Це рішення опиралося на те, що у силу своєї надлишковості файли цього формату добре підходять для реалізації методу LSB.

Щоб визначити область вбудування повідомлень потрібно розглянути структуру wav-файлів. Канонічна структура представлена на рис. 2.2.

	Найменування	Розмір (байт)	Значення
0	ChunkID	4	"RIFF" (0x52494646)
4	ChunkSize	4	Розмір файлу - 8
8	Format	4	"WAVE" (0x57415645)
12	Subchunk1ID	4	"fmt " (0x666D7420)
16	Subchunk1Size	4	16
20	CompressCode	2	1 - 65535
22	NumChannels	2	1 - 65535
24	SampleRate	4	1 - 0xFFFFFFFF
28	ByteRate	4	1 - 0xFFFFFFFF
32	BlockAlign	2	1 - 65535
34	BitsPerSample	2	2 - 65535
36	Subchunk2ID	4	"data" (0x64617461)
40	Subchunk2Size	4	1 - 0xFFFFFFFF
44	Data	Subchunk 2Size	

Рисунок 2.2 - Структура wav-файлу

Файли формату wav використовують стандартну структуру RIFF, що групує вміст файлів в окремі секції, кожен з яких має заголовок та розмір секції [7].

Перші 8 байт являють собою стандартний заголовок секції RIFF. Він складається з назви заголовку та розміру секції, що дорівнює розміру файлу – 8

байт, які були використані для заголовку. Наступні 4 байти описують тип ресурсу, який у wav-файлів завжди дорівнює «WAVE».

Найпростіші wav-файли мають лише 2 підсекції:

- Підсекція формату «fmt »;
- Підсекція даних «data».

Ці підсекції є обов'язковими, так як містять у собі опис формату вибірок аудіоданих та самі аудіодані.

Детальна інформація щодо структури підсекцій «fmt » та «data» наведена у табл. 2.1 та табл. 2.2 відповідно.

Таблиця 2.1 - Структура підсекції формату

Розмір (байт)	Ім'я	Опис	Значення
4	Chunk ID	ID секції	"fmt " (0x666D7420)
4	Chunk Data Size	Розмір даних секції	16 + розмір додаткових даних формату
2	Compression Code	Код типу стиснення звукових даних	1 - 65 535
2	Number of channels	Кількість каналів	1 - 65 535
4	Sample rate	Частота дискретизації	1 - 0xFFFFFFFF
4	Average bytes per second	Кількість байт на секунду	1 - 0xFFFFFFFF
2	Block align	Розмір блоку	1 - 65535
2	Significant bits per sample	Кількість значущих біт на вибірку	2 - 65 535
2	Extra format bytes	Розмір додаткових даних формату	0 - 65 535
		Додаткові дані формату	

Таблиця 2.2 - Структура підсекції даних

Розмір (байт)	Назва	Опис	Значення
4	Chunk ID	ID секції	"data" (0x64617461)
4	Chunk Data Size	Розмір даних секції	1 - 0xFFFFFFFF
	Data	Дані	

Приховування секретного повідомлення у wav-контейнер методом найменшого значущого біта здійснюється лише у підсекції даних після Chunk ID та Chunk Data Size. Також для вкраплення повідомлення так, щоб слухова система людини нічого не запідозрила, потрібно звернути увагу на кількість значущих біт на вибірку, що указані у підсекції формату. Це значення вказує кількість біт, що формують кожну вибірку сигналу, саме воно використовується для визначення проміжку, через який можна приховувати інформацію методом найменшого значущого біта.

2.3 Структура службового заголовку

Для того, щоб коректно виконувати вилучення секретного повідомлення з заповненого стегоконтейнеру, при приховуванні повідомлення також потрібно записати у файл свій заголовок. У конкретному випадку він представляє послідовність з 8 байтів. Заголовок наведений на рис. 2.3.

Рисунок 2.3 - Заголовок для приховуваного повідомлення

Де:

- 0 – 31 біт – розширення файлу таємного повідомлення;
- 32-й біт – наявність шифрування;
- 33 – 63 біт – розмір повідомлення.

Перші чотири байти відводяться на запис розширення файлу, що приховується. 32 біт відповідає за наявність або відсутність шифрування. Все інше місце відводиться на запис розміру файлу, що приховується.

2.4 Висновки за розділом

У даному розділі наведено основні функції комплексу, що розробляється. Для посилення захисту інформації прийнято рішення реалізувати шифрування AES 128. У якості звукового контейнеру ухвалено використовувати файли формату «wav», розглянуто його структуру для подальшого коректного приховування повідомлення.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір середовища та засобів розробки

Для реалізації програмного комплексу аудіостеганографії обрано середовище візуального програмування Microsoft Visual Studio 2019 [8]. Це 64-розрядне інтегроване середовище розробки спрощує роботу з великими проектами та складними робочими навантаженнями. При виконанні справ, таких як написання коду та перемикання гілок, система реагує швидко та плавно. Також у середовищі присутній потужний набір засобів автоматичного завершення коду IntelliCode, які розпізнають контекст коду, а саме імена змінних, функції та тип створюваного коду. Це дозволяє IntelliCode відразу завершувати цілий рядок, допомагаючи впевненіше та точніше створювати код.

У якості мови програмування використовується C# [9]. Це сучасна об'єктно-орієнтована і типобезпечна мова програмування. C# дозволяє розробникам створювати різні типи безпечних та надійних програм, що виконуються в .NET. Також мова реалізує функції, що дозволяють створювати надійні та стійкі програми, такі як:

- Складання сміття, що автоматично звільняє пам'ять, зайняту недосяжними об'єктами, що не використовуються;
- Типи можуть допускати значення null, тим самим забезпечуючи захист від змінних, які не посилаються на виділені об'єкти;
- Обробка винятків, що надає структурований та розширюваний підхід до виявлення помилок та відновлення після них.

Платформою інтерфейсу користувача обрано Windows Presentation Foundation (WPF), яка є частиною платформи .NET [10]. Для візуалізації використовується DirectX (апаратна підтримка), у випадку з старими відеокартами використовується програмне обчислення об'єктів. WPF візуалізує всі елементи інтерфейсу користвача самостійно, в основі масштабування ставиться системний параметр Dots Per Inch (DPI). WPF використовує

розширювану мову розмітки додатків (XAML), щоб надати декларативну модель для програмування додатків. Нижче представлені основи XAML:

- Кожен елемент у документі XAML відображається як екземпляр класу .NET. Ім’я елементу відповідає імені класу;
- XAML допускає вкладення одного елементу в інший;
- Властивості кожного класу можна встановлювати через атрибути.

Для реалізації шифрування алгоритмом AES 128 у програмі використовується стандартний клас Aes [11].

3.2 Програмна реалізація режиму приховування

Режим приховування залежить від значення кількості значущих біт на вибірку. Програмний комплекс реалізує приховування при 8, 16, 24 та 32 біт на вибірку. Якщо кількість значущих біт на вибірку дорівнює 8, у кожну вибірку заноситься 2 біти секретного повідомлення, якщо кількість дорівнює 16, у кожну вибірку заноситься 4 біти секретного повідомлення, якщо ж кількість дорівнює 24 або 32, у кожну вибірку заноситься 1 байт секретного повідомлення.

Блок-схема узагальненого алгоритму приховування повідомлення у стегоконтейнер представлена на рис. 3.1. Реалізація мовою C# представлена у Додатку Б у методі Encrypt().

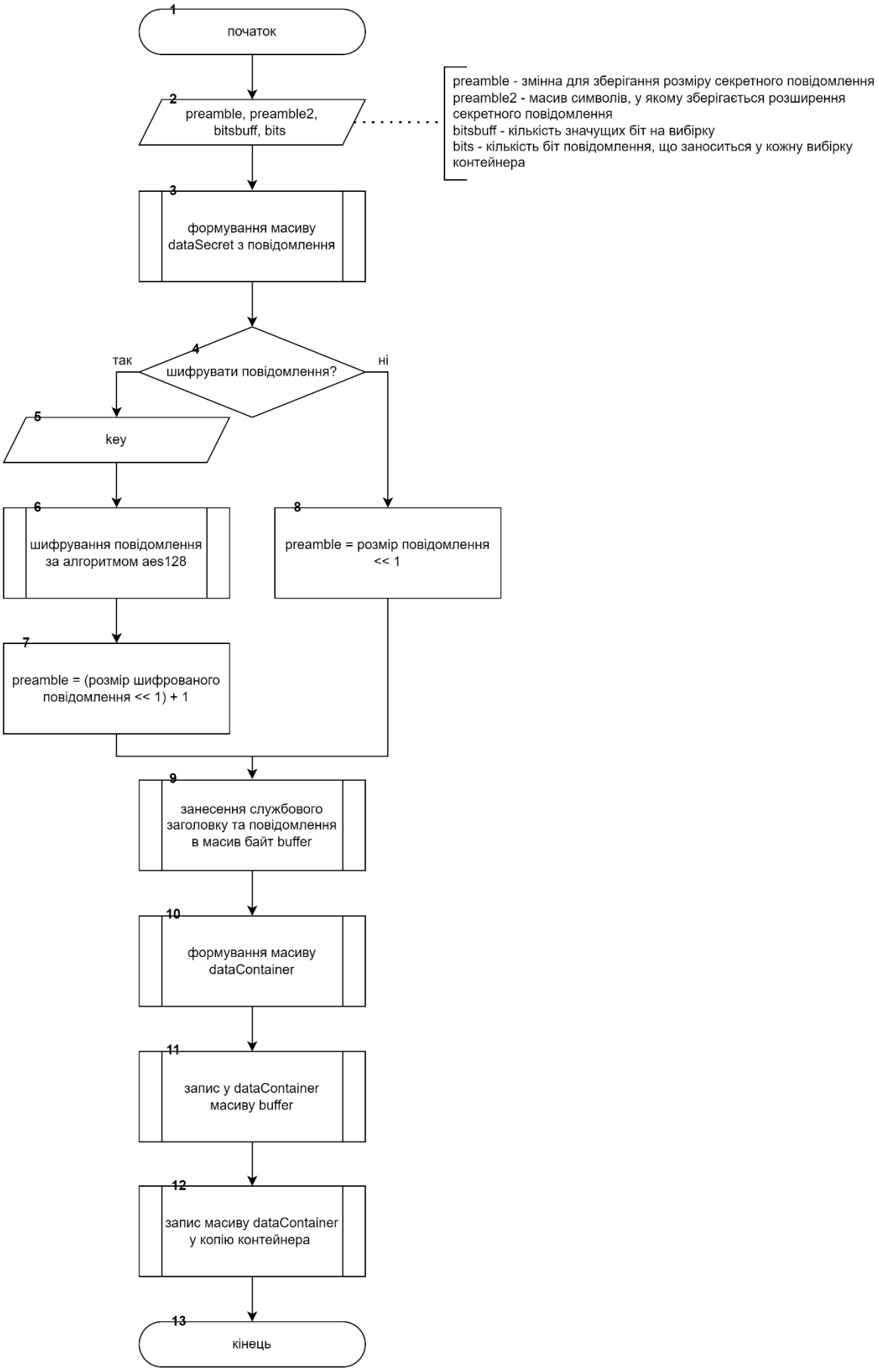

Рисунок 3.1 – Блок-схема узагальненого алгоритму приховування повідомлення

У другому блоці оголошується змінна для зберігання довжини секретного повідомлення та біту наявності шифрування preamble, масив символів, у якому зберігається розширення файлу секретного повідомлення preamble2. Оголошується змінна bitsbuff, у яку заноситься кількість значущих біт на вибірку, котра зчитується з заголовку контейнеру, що обрав користувач. Оголошується змінна bits, яка дорівнює 2, 4 або 8, у залежності від кількості значущих біт на вибірку.

У третьому блоці формується масив dataSecret типу byte. Його розмірність дорівнює довжині повідомлення. Після цього повідомлення побайтно записується у створений масив dataSecret.

У четвертому – восьмому блоках проводиться перевірка наявності шифрування. Якщо шифрування наявне, програмний комплекс зчитує шістнадцяти символьний ключ введений користувачем та передає його у масив key. Після цього масиви dataSecret та key передаються до методу шифрування та у результаті масив dataSecret перезаписується на зашифроване повідомлення. У змінну preamble записується довжина масиву dataSecret зрушена на один біт вліво та в нульовий біт записується одиниця у знак того, що повідомлення зашифроване. Якщо ж шифрування відсутнє, то у змінну preamble записується довжина масиву dataSecret зрушена на один біт вліво.

У дев’ятому блоці створюється масив buffer типу byte розмірністю, що дорівнює:

$$\text{Довжина масиву } dataSecret * \frac{8}{bits} + \frac{64}{bits}$$

Після цього масив buffer заповнюється змінною preamble, масивом символів preamble2 та масивом dataSecret. У кожну комірку масиву buffer заноситься по bits біт інформації.

У десятому блоці створюється копія файлу пустого контейнеру, після чого він зчитується з місця закінчення заголовку підсекції data та побайтно записується у новий масив dataContainer.

У одинадцятому блоці у молодші біти кожної вибірки створеної копії контейнеру замінюються на елементи масиву buffer. У елементах масиву dataContainer, порядковий номер яких кратний $\frac{bitsbuff}{8}$, замінюються молодші біти кількістю bits на елементи масиву buffer.

У дванадцятому блоці масив dataContainer записується у копію пустого контейнеру з місця закінчення заголовку підсекції data.

3.3 Програмна реалізація режиму вилучення

Блок-схема узагальненого алгоритму вилучення повідомлення у стегоконтейнер представлена на рис. 3.2. Реалізація мовою C# представлена у Додатку Б у методі Decrypt().

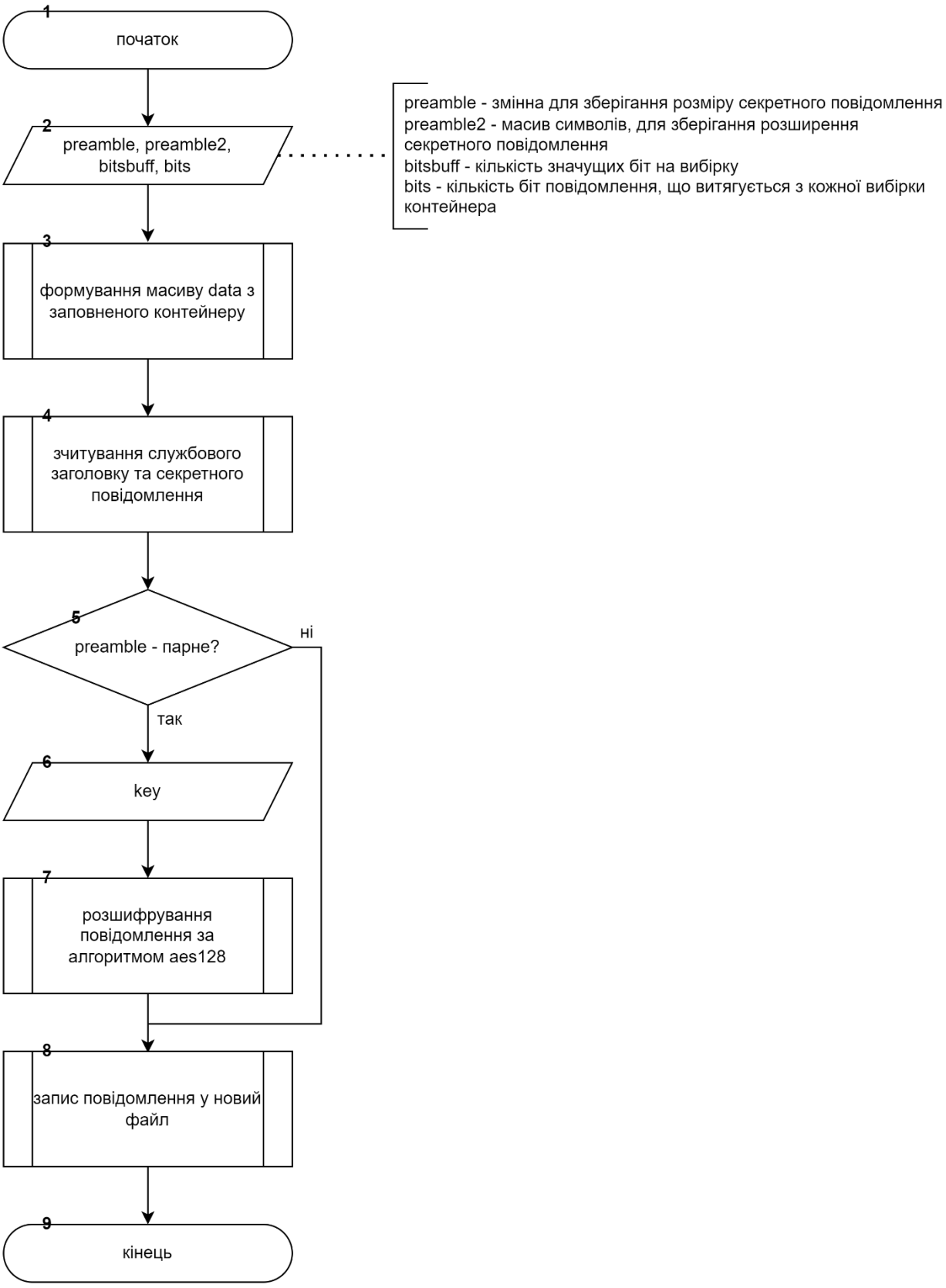

Рисунок 3.2 – Блок-схема узагальненого алгоритму вилучення повідомлення

У другому блоці оголошується змінна для зберігання довжини секретного повідомлення та біту наявності шифрування preamble, масив символів для

зберігання розширення файлу секретного повідомлення preamble2. Оголошується змінна bitsbuff, у яку заноситься кількість значущих біт на вибірку, котра зчитується з заголовку контейнеру, що обрав користувач. Оголошується змінна bits, яка дорівнює 2, 4 або 8, у залежності від кількості значущих біт на вибірку.

У третьому блоці побайтно зчитується заповнений контейнер та на його основі створюється масив data.

У четвертому блоці з масиву data з місця закінчення заголовку підсекції data контейнера зчитується:

- довжина прихованого повідомлення та біт наявності шифрування, що заносяться у змінну preamble;
- розширення файлу секретного повідомлення, що заноситься у масив символів preamble2;
- саме секретне повідомлення, що заноситься у масив dataSecret типу byte розмірністю, яка дорівнює довжині прихованого повідомлення.

Зчитування відбувається по bits біт з кожної вибірки контейнера.

У п’ятому блоці проводиться перевірка наявності шифрування секретного повідомлення. Якщо нульовий біт preamble дорівнює одиниці, то шифрування наявне, якщо ж нулю, то відсутнє.

Якщо шифрування наявне, то додатково виконуються шостий та сьомий блоки. Користувач вводить шістнадцяти символьний ключ, який заноситься у масив key типу byte і після цього разом з масивом зашифрованих даних dataSecret передається у метод розшифрування. Результатом отримаємо перезаписаний масив dataSecret, у якому зберігаються розшифроване повідомлення.

У восьмому блоці створюється новий файл з розширенням preamble2, у який записується дані масиву dataSecret.

3.4 Перевірка працездатності програмного комплексу

При взаємодії користувача з інтерфейсом програмного комплексу обирається контейнер та секретне повідомлення. Для перевірки у якості секретного повідомлення було взято файл, що наведений на рис. 3.3. На виході користувач отримує заповнений контейнер, що ідентичний за розміром (див. рис. 3.4) та за звучанням (для людського слуху) з пустим контейнером.

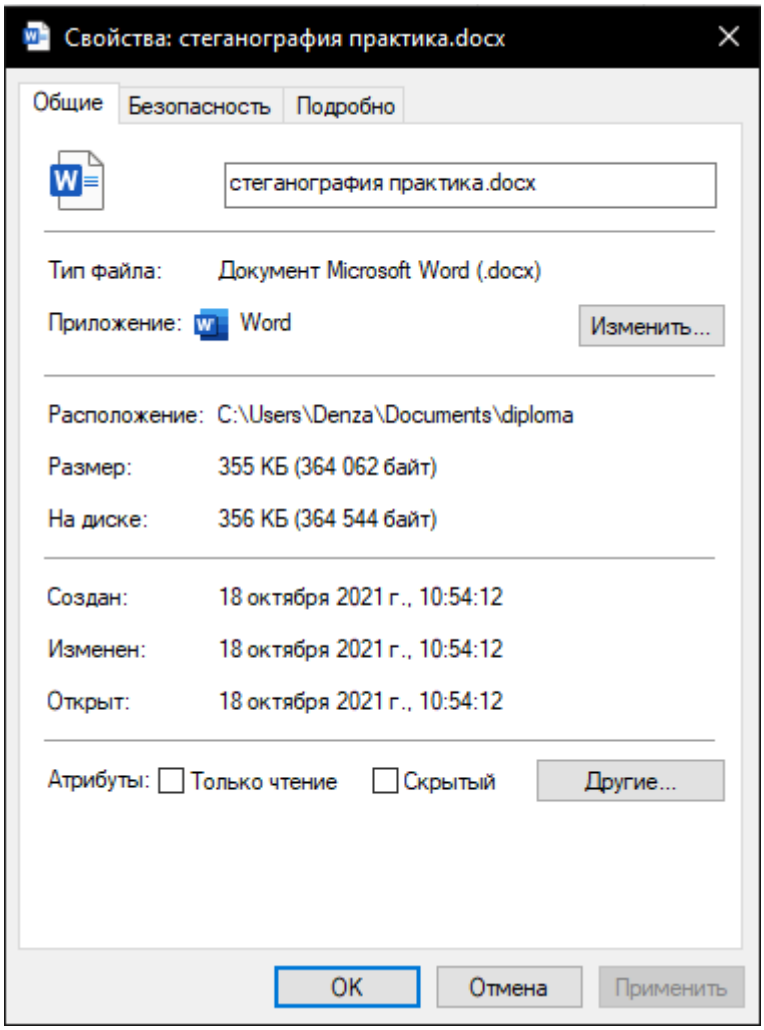

Рисунок 3.3 - Секретне повідомлення

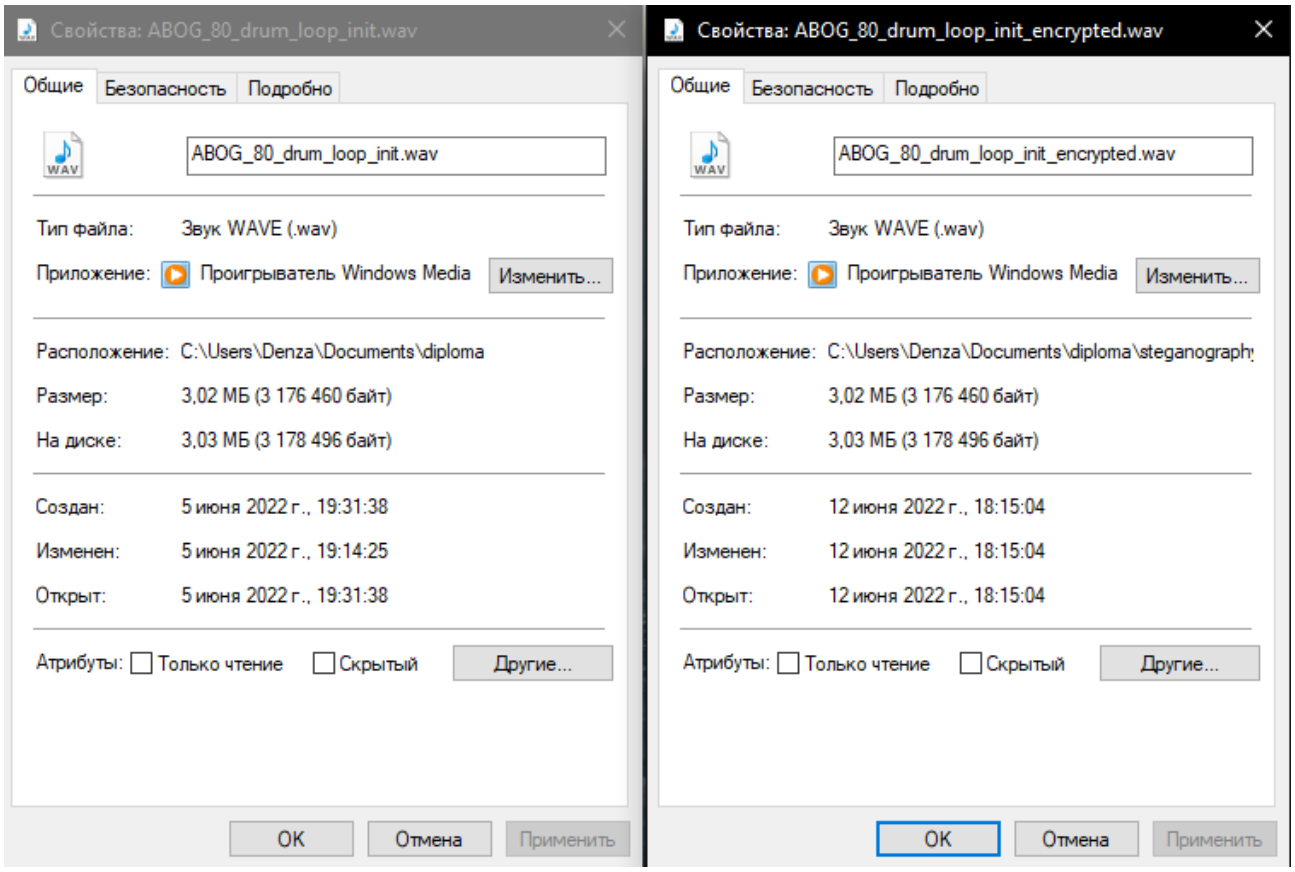

Рисунок 3.4 - Відповідність розмірів пустого та заповненого контейнеру

На рис. 3.5 – 3.6 представлені осцилограми пустого та заповненого контейнеру відповідно. Різниці між ними практично не видно.

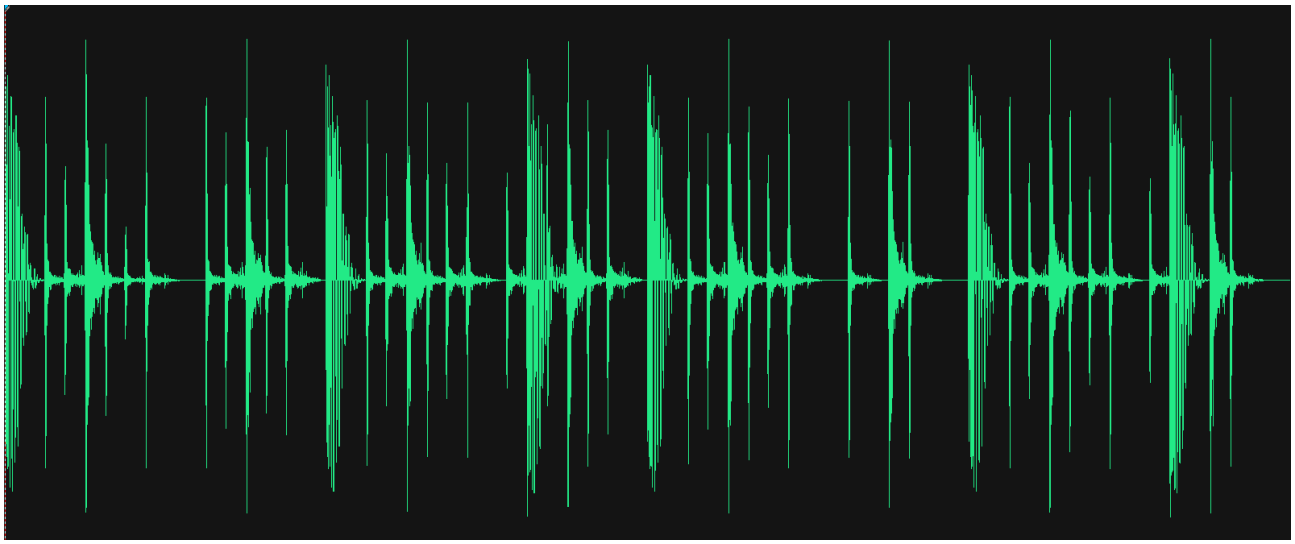

Рисунок 3.5 - Осцилограма пустого контейнеру

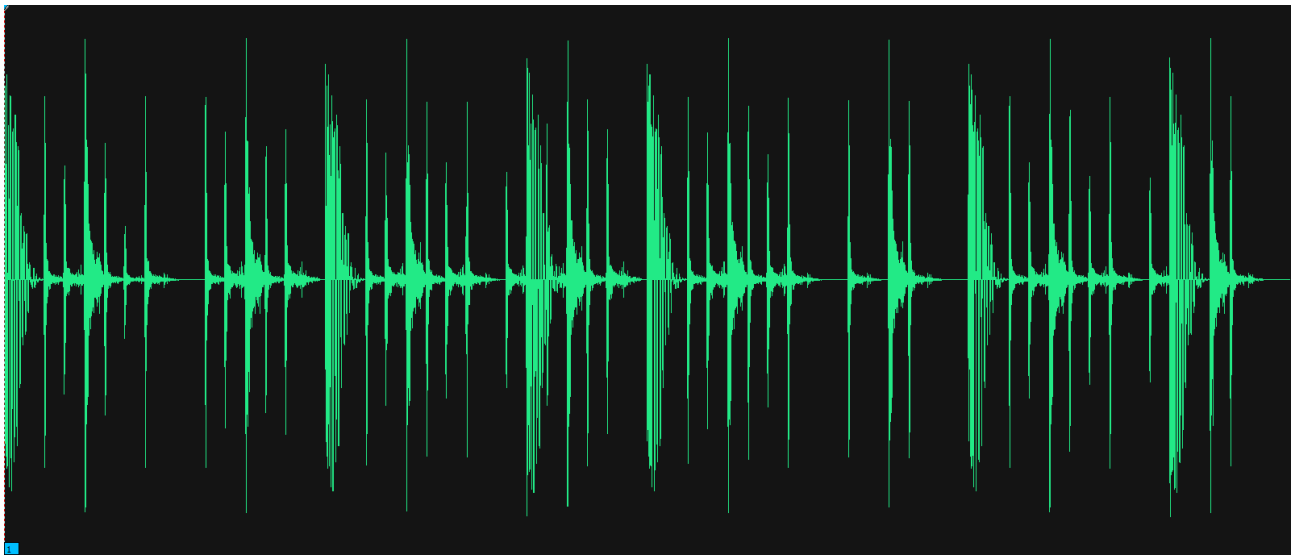

Рисунок 3.6 - Осцилограма заповненого контейнеру

За допомогою додатку Hex Editor Neo [12] розглянемо заповнений контейнер у вигляді байтів (див. рис. 3.7).

ABOG_80_drum_loop_init_encrypted.wav

00000059	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	52	49	46	46	04	78	30	00	57	41	56	45	4a	55	4e	4b	RIFF.x0.WAVEJUNK
00000010	1c	00	00	00	04	78	30	00	00	00	00	00	20	73	30	00	.....x0..... s0.
00000020	00	00	00	00	30	13	08	00	00	00	00	00	00	00	00	00	....0...........
00000030	66	6d	74	20	10	00	00	00	01	00	02	00	44	ac	00	00	fmt ....D¬..
00000040	98	09	04	00	06	00	18	00	64	61	74	61	20	73	30	00	....data s0.
00000059	00	00	00	0b	00	00	1c	ff	ff	3c	ff	ff	64	ff	ff	6f	[illegible]
00000060	ff	ff	63	ff	ff	78	ff	ff	50	00	00	4b	ff	ff	03	00	яяcяяxяяP..Кяя..
00000070	00	04	00	00	14	ff	ff	00	00	00	06	ff	ff	00	00	00	.....яя....яя...
00000080	08	ff	ff	00	ff	ff	00	00	00	00	ff	ff	21	ff	ff	00	.яя.яя....яя!яя.
00000090	ff	ff	bf	ff	ff	6b	ff	ff	ad	ff	ff	3e	ff	ff	78	ff	яяїяякяя-яя>яяхя
000000a0	ff	01	ff	ff	00	ff	ff	00	00	00	09	ff	ff	06	ff	ff	я.яя.яя....яя.яя
000000b0	00	ff	ff	00	ff	ff	13	ff	ff	00	ff	ff	08	00	00	02	.яя.яя.яя.яя....
000000c0	00	00	5b	fe	ff	43	ff	ff	6f	fc	ff	6e	ff	ff	74	fc	..[юяСяяоьяпяять
000000d0	ff	65	00	00	6e	f3	ff	74	f7	ff	5f	f4	ff	54	f8	ff	яе..nуятчя_фяТшя
000000e0	79	08	00	70	0b	00	65	04	00	73	09	00	5d	00	00	2e	у..p..e..s..]...
000000f0	07	00	78	02	00	6d	08	00	6c	04	00	20	08	00	a2	fe	..x..m..l.. ..ўю
00000100	ff	04	02	00	02	f8	ff	28	fd	ff	a0	1d	00	00	23	00	я....шя(эя ...#.
00000110	02	21	00	00	26	00	00	0f	00	00	11	00	00	61	00	00	.!..&.........a..
00000120	65	00	00	15	00	00	1c	00	00	00	ff	00	06	ff	00	73	e.........я..я.s
00000130	00	00	77	00	00	09	03	00	0c	03	00	3a	00	00	3f	00	..w.........:..?.
00000140	00	d1	fa	00	d6	fa	00	d7	fb	00	db	fb	00	78	02	00	.Съ.Цъ.Чы.Ыы.х..
00000150	7b	02	00	b9	06	00	bd	06	00	a1	05	00	a4	05	00	72	{..№..S..Ў..¤..r
00000160	01	00	75	01	00	07	fd	00	0a	fd	00	c7	fb	00	ca	fb	..u...э..э.Зы.Кы
00000170	00	29	fb	00	2c	fb	00	d0	f9	00	d5	f9	00	05	fd	00	.)ы.,ы.Рщ.Хщ..э.
00000180	0b	fd	00	9f	04	00	a4	04	00	82	06	00	89	06	00	93	.э.ц..¤..,..‰..“

Рисунок 3.7 – Фрагмент заповненого контейнеру

Червоним виділено кількість значущих біт на одну вибірку, яка дорівнює 18 у шістнадцятирічному або 24 у десятирічному форматі числа. Для такої вибірки у програмі реалізовано приховування по 8 біт інформації на вибірку.

Синім виділено службовий заголовок, що складається з довжини секретного повідомлення, біту наявності шифрування та розширення файлу секретного повідомлення. Тобто довжина повідомлення з бітом наявності шифрування разом дорівнюють 0xb1c3c, або 0b1011 0001 1100 0011 1100. Біт шифрування дорівнює 0, тому шифрування відсутнє, а довжина повідомлення дорівнює 0b0101 1000 1110 0001 1110 або 364062 байти. Розширення файлу секретного повідомлення ж дорівнює 0x646f6378, або «docx».

Чорним виділено чотири байти, що повністю відповідають першим чотирьом байтам секретного повідомлення (див. рис. 3.8).

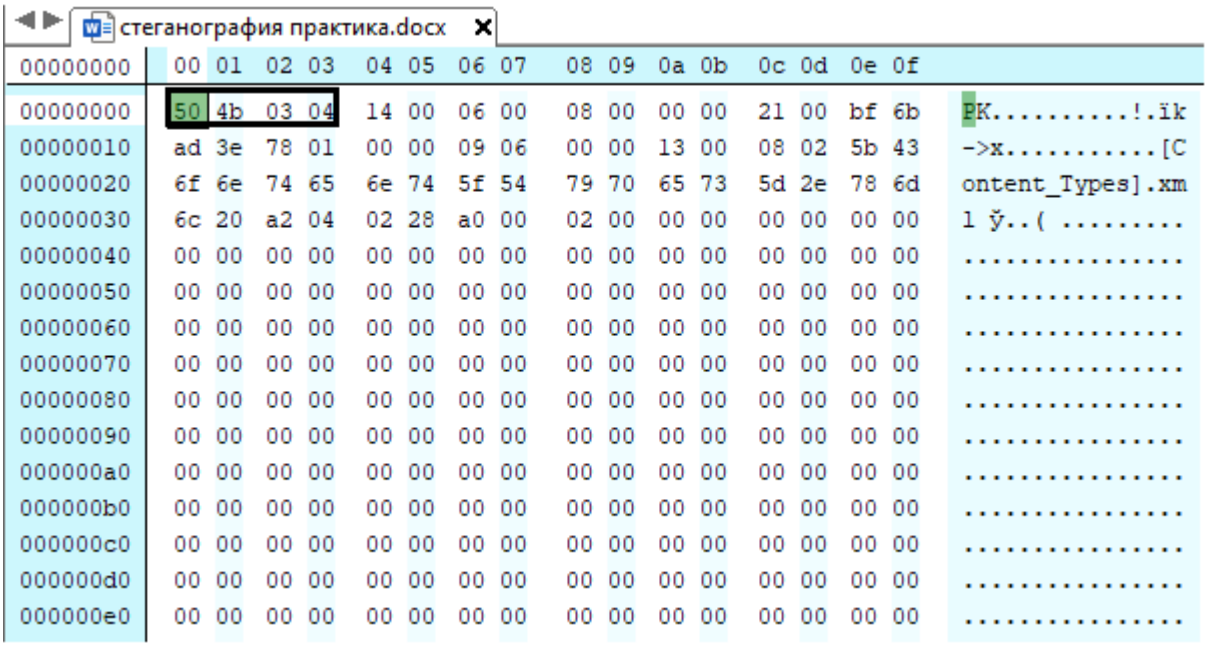

Рисунок 3.8 - Фрагмент секретного повідомлення

Результатом вилучення секретного повідомлення з заповненого повідомлення є файл, повністю ідентичний початковому секретному повідомленню (див. рис. 3.9 – 3.10).

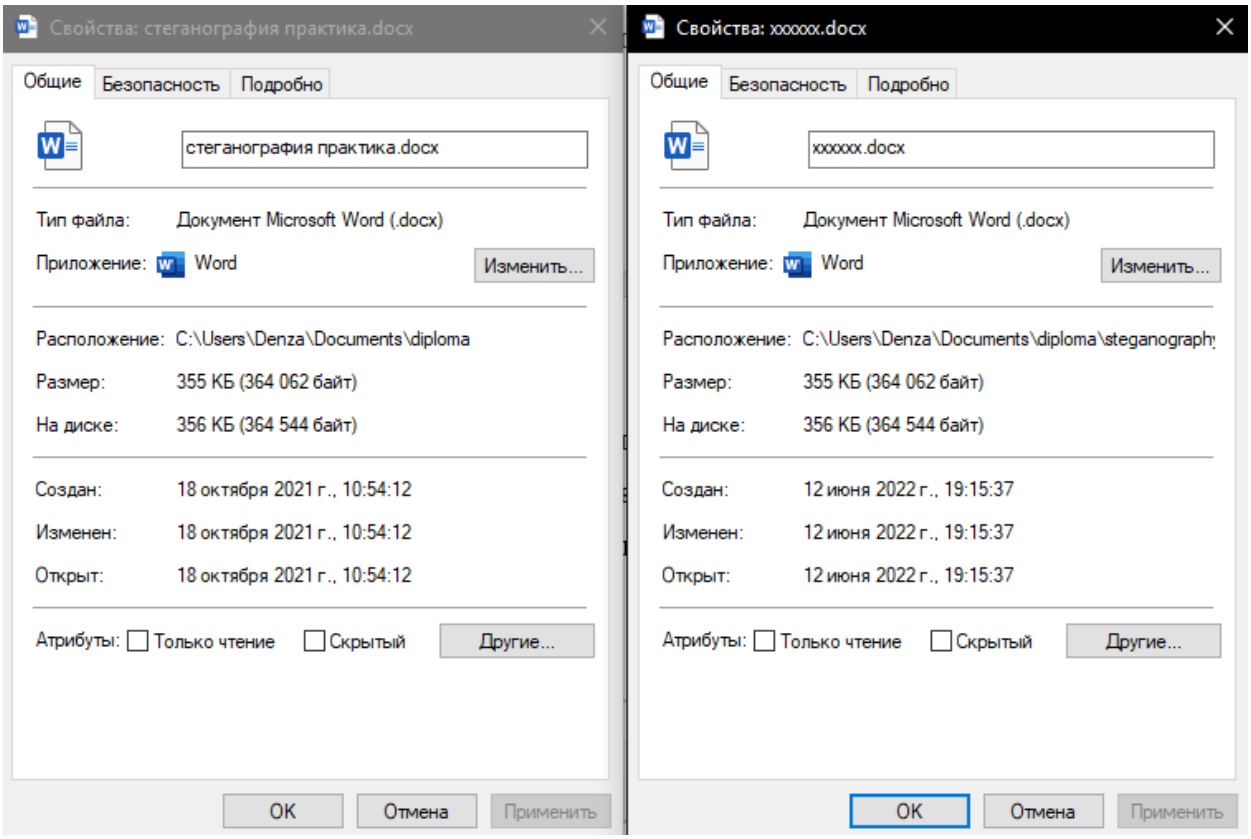

Рисунок 3.9 - Відповідність розмірів секретного повідомлення та повідомлення, вилученого з заповненого контейнеру

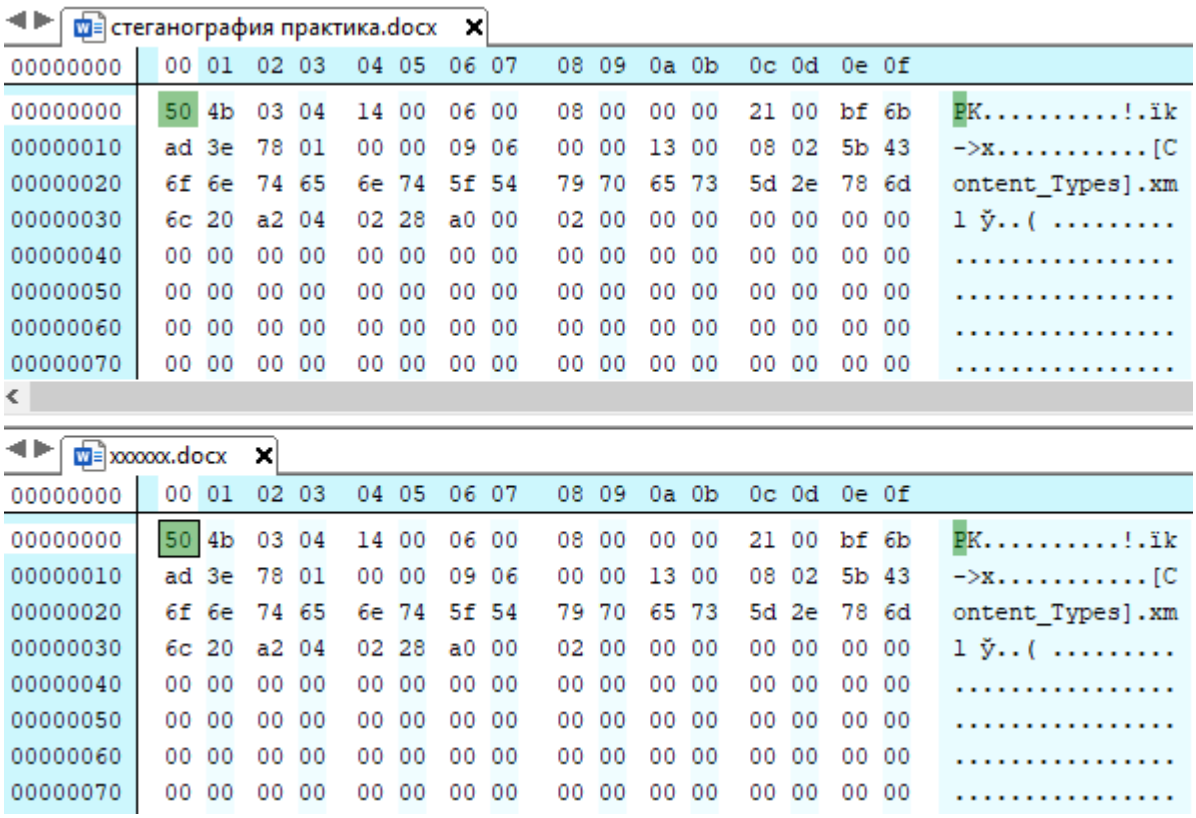

Рисунок 3.10 - Фрагмент секретного повідомлення та повідомлення, вилученого з заповненого контейнеру

Програмний комплекс також був перевірений при наявності шифрування, та з контейнерами, у яких кількість значущих біт дорівнювала 16 та 32. Програма працює справно.

3.5 Інструкція з використання програмного комплексу

Розроблюваний програмний комплекс може використовуватися як засіб стеганографічного захисту інформації. Після запуску програми користувача зустрічає головне вікно, що представлено на рис. 3.11.

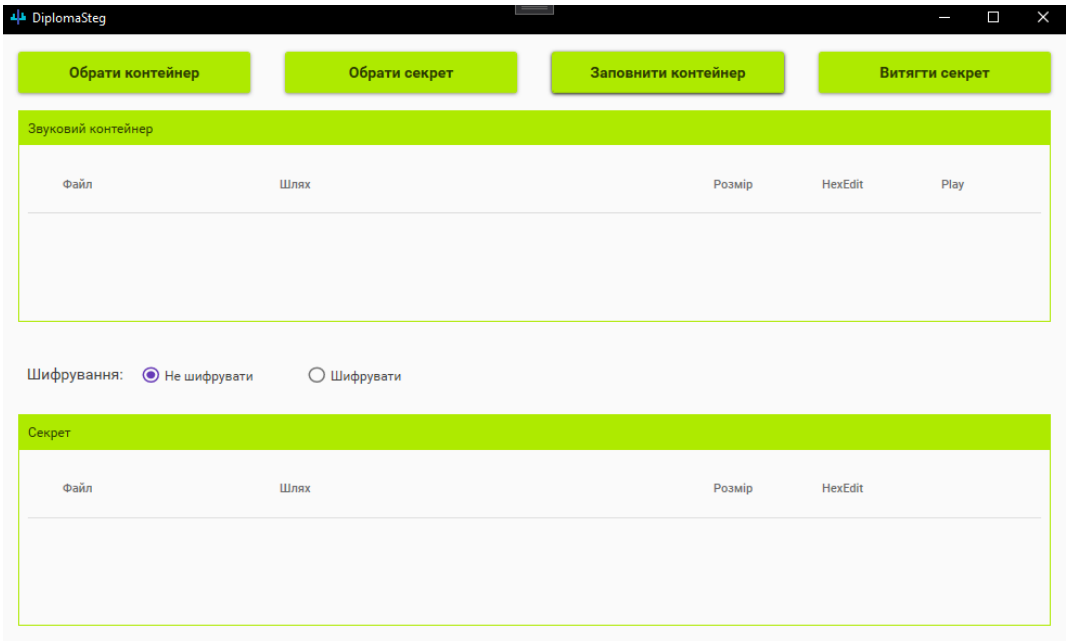

Рисунок 3.11 - Головне вікно програми

Щоб було наочніше, вирішено обрати контейнер з секретним повідомленням та пронумеровати важливі складові програмного комплексу (див. рис. 3.12).

Рисунок 3.12 – Головне вікно програми з обраним контейнером та секретним повідомленням

Щоб сховати секретне повідомлення у стегоконтейнер, користувачеві потрібно виконати ряд дій, такі як вибір контейнеру та секретного повідомлення, вибір варіанту шифрування та саме приховання повідомлення.

Обрати контейнер. Користувач повинен натиснути на кнопку 1 «Обрати контейнер», програмним додатком буде виконана дія відкриття провідника з фільтром для можливості обирання файлів тільки формату «wav». Після того як контейнер обрано, його назва, шлях до нього, розмір, поле 6, що вказує на розмір повідомлення, що можна сховати у стегоконтейнері, кнопки 7 та 8 з’являться на головному інтерфейсі. Контейнер можна відкрити у додатку «Hex Editor Neo», натиснувши кнопку 7 та прослухати, натиснувши на кнопку 8.

Обрати секретне повідомлення. Користувач повинен натиснути на кнопку 2 «Обрати секрет», після чого обирає файл. Коли обрано секретне повідомлення, як у випадку із контейнером, назва секретного повідомлення, шлях до нього, розмір та кнопка 7 з’явиться на головному інтерфейсі. Секретне повідомлення також можна відкрити у додатку «Hex Editor Neo».

Наступним кроком користувач має можливість у блоці 5 обрати варіант шифрування секретного повідомлення, після чого вже сховати повідомлення у контейнер, натиснувши на кнопку 3 «Заповнити контейнер». Якщо користувач обрав секретне повідомлення, що є більшим за розмір вільного місця у контейнері, з’явиться вікно з помилкою (див. рис. 3.13) та приховування не відбудеться.

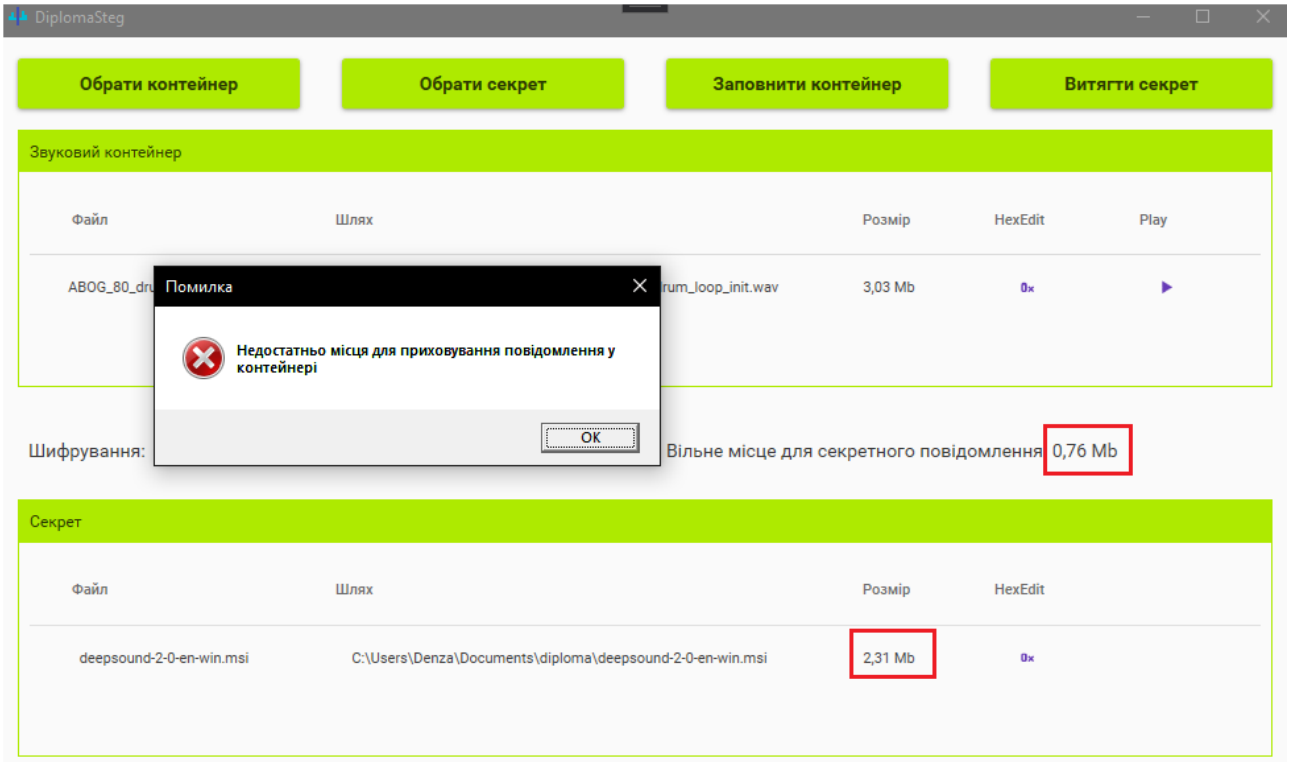

Рисунок 3.13 - Невдале приховування повідомлення

Якщо користувач обрав шифрувати файл, то при приховуванні секретного повідомлення з’явиться вікно, що представлене на рис. 3.14. Користувач повинен бути записати пароль з 16 символів, в іншому випадку приховування повідомлення не відбудеться (див. рис. 3.15 та 3.16).

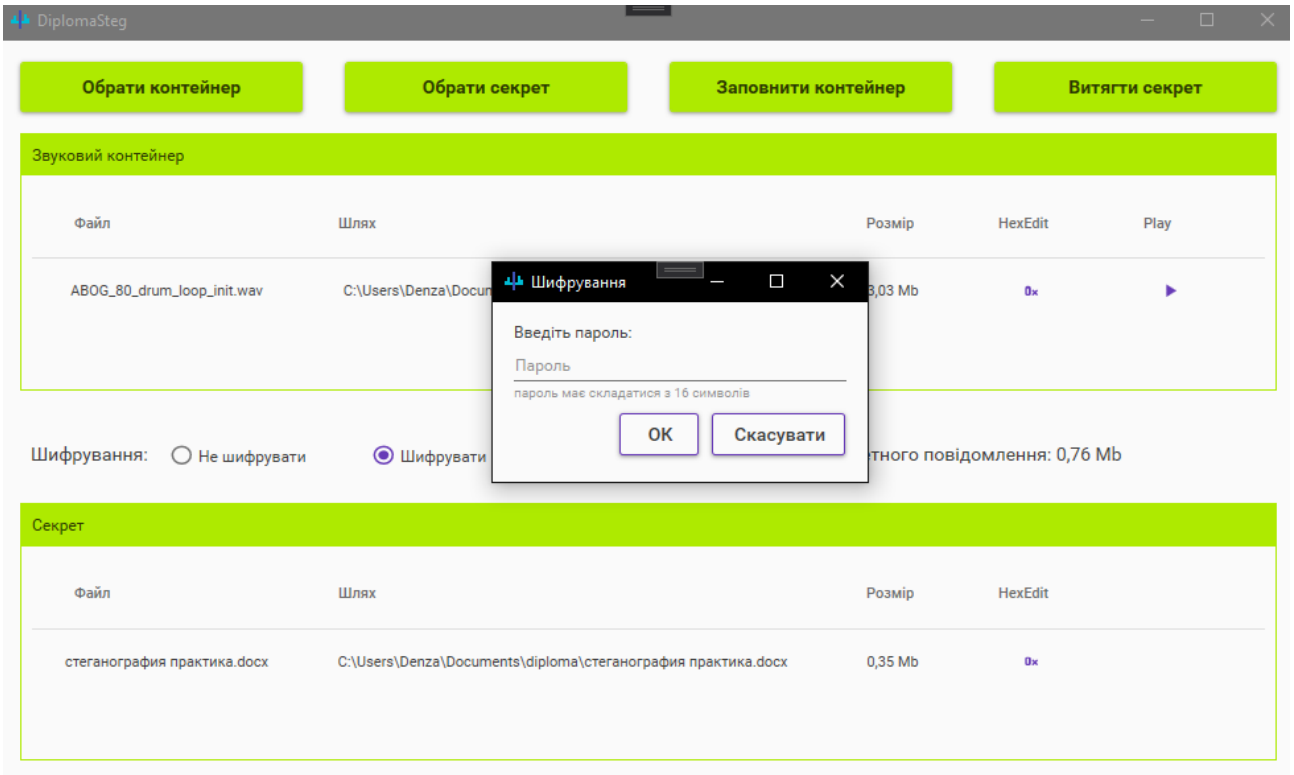

Рисунок 3.14 - Вікно введення паролю

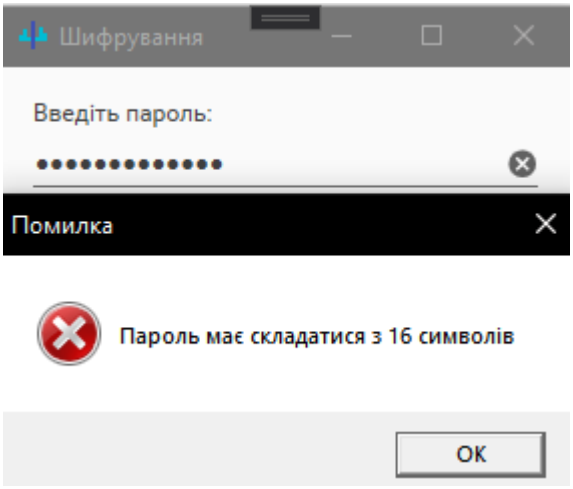

Рисунок 3.15 - Вікно помилки при не коректному записі паролю

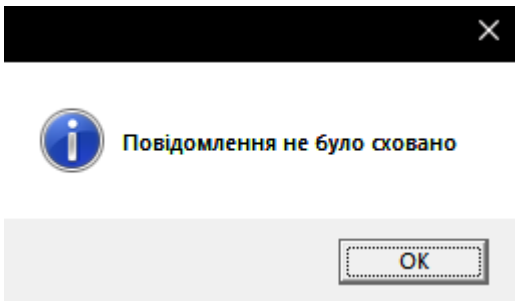

Рисунок 3.16 - Вікно повідомлення при скасуванні шифрування

Щоб витягти секретне повідомлення з заповненого стегоконтейнеру, користувач обирає контейнер та натискає на кнопку 4 «Витягти секрет». Якщо

приховане повідомлення зашифроване, користувач повинен бути ввести коректний пароль для його розшифрування. Якщо користувач невірно введе пароль, у результаті він отримає бракований файл, що не зчитується.

3.6 Висновки за розділом

У даному розділі обрано середовище розробки візуального програмування Microsoft Visual Studio 2019, мовою розробки обрано C#. Для створення графічного інтерфейсу обрано Windows Presentation Foundation (WPF). Розроблено програмне забезпечення комплексу в режимах приховування та вилучення таємного повідомлення, створені відповідні блок-схеми. Перевірена працездатність програмного забезпечення та написана інструкція з його використання.

ВИСНОВКИ

У даній роботі розроблено програмний комплекс стеганографічного захисту інформації з використанням звукових файлів-контейнерів формату «wav» за методом LSB.

Для виконання поставленої мети у роботі наведені визначення основних понять сучасної стеганографії, виконаний огляд відомих програмних засобів аудіостеганографії, а також розглянуті та проаналізовані методи цифрової стеганографії для звукових контейнерів. Результатом аналізу стало рішення обрати метод LSB для реалізації у комплексі. Його обрано тому, що серед розглянутих методів приховування інформації він найпростіший у реалізації, має великий обсяг для приховування повідомлення порівняно з іншими методами, та відносно непомітність на слух.

На основі першого етапу роботи сформований перелік основних функцій програмного комплексу, вирішено реалізувати шифрування AES 128 для посилення захисту інформації. Також у якості звукового контейнеру вирішено використовувати файли формату «wav» та розглянуто його структуру.

Обрано середовище розробки візуального програмування, мову та платформу для реалізації інтерфейсу користувача, розроблені алгоритми приховування та вилучення таємного повідомлення у вигляді блок-схем та розроблено відповідне програмне забезпечення. Перевірена працездатність програми та написана інструкція з її використання.

Розроблене програмне забезпечення може бути використано як в практичних цілях для приховування інформації, так і в навчальному процесі студентів відповідних спеціальностей при проведенні лабораторних або практичних робіт.

ПЕРЕЛІК ПОСИЛАНЬ

1. What is steganography? [Електронний ресурс] – Режим доступу: https://www.techtarget.com/searchsecurity/definition/steganography
2. Основные понятия и определения стеганографии [Електронний ресурс]. – Режим доступу: https://studref.com/608317/informatika/osnovnye_ponyatiya_opredeleniya_steganografii
3. R. Sridevi, A. Damodaram, S.V.L. Narasimham, “Efficient Method of Audio Steganography by Modified LSB Algorithm and Strong Encryption Key with Enhanced Security”, Journal of Theoretical and Applied Information Technology, vol. 5, no. 6, pp. 768 – 771, 2009.
4. W. Bender, D. Gruhl, N. Morimoto, “Techniques for Data Hiding”, IBM Systems Journal, vol. 35, no. 3, pp. 313 – 336, 1996.
5. D. Huang, T. Yeo, “Robust and Inaudible Multi-echo Audio Watermarking”, Proceedings of the IEEE Pacific-Rim Conference on Multimedia, pp. 615 – 622, Taipei, China, 2002.
6. D. Kirovski, H. Malvar, “Spread spectrum Watermarking of Audio Signals”, IEEE Transactions on Signal Processing, vol. 51, no. 4, pp. 1020 – 1033, 2003.
7. Wav file format [Електронний ресурс] – Режим доступу: https://sites.google.com/site/musicgapi/technical-documents/wav-file-format
8. Code faster. Work smarter [Електронний ресурс] – Режим доступу: https://visualstudio.microsoft.com/vs/
9. A tour of the C# language [Електронний ресурс] – Режим доступу: https://docs.microsoft.com/en-gb/dotnet/csharp/tour-of-csharp/
10. WPF overview [Електронний ресурс] – Режим доступу: https://docs.microsoft.com/en-gb/dotnet/desktop/wpf/introduction-to-wpf?view=netframeworkdesktop-4.8
11. Aes Class [Електронний ресурс] – Режим доступу: https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.aes?view=net-6.0

12. Free Hex Editor Neo [Електронний ресурс] – Режим доступу: https://www.hhdsoftware.com/free-hex-editor