Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Чураков Іван Миколайович,
студент групи РЗ-181

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Розробка стеганографічного алгоритму приховування даних в аудіо сигналах

Спеціальність:
125 Кібербезпека

Спеціалізація, освітня програма:
Кібербезпека

Керівник:
Трифонова Катерина Олексіївна,
ст. викладач

Одеса – 2022

Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій
Кафедра кібербезпеки та програмного забезпечення

Рівень вищої освіти: перший (бакалаврський)
Спеціальність 125 – Кібербезпека
Освітня програма – Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри КБПЗ
д.т.н.,проф. А.А. Кобозєва
_____
«____»_____20__р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
Чуракову Івану Миколайовичу

1. Тема роботи: *Розробка стеганографічного алгоритму приховування даних в аудіо сигналах*,
керівник роботи: *ст. викл. Трифонова Катерина Олексіївна,*
затверджені наказом ректора № 168-в від 17.05.2022р.
2. Зміст роботи: *аналіз стеганографічних алгоритмів для цифрового аудіо сигналу; теоретичні основи модифікації стеганографічного алгоритму; практична реалізація стеганографічного алгоритму для цифрового аудіо сигналу*
3. Перелік ілюстративного матеріалу: *класифікація стеганографічних методів для аудіо файлів; блок-схема алгоритму роботи стеганографічної програми*

4. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	доц. Ярова І.А.		

5. Дата видачі завдання «____»_____ 20___ р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання	Примітка
1	Огляд стеганографічних алгоритмів для цифрового аудіо сигналу	21.03.22	виконано
2	Модифікація стеганографічного алгоритму для цифрового аудіо сигналу	04.04.22	виконано
3	Реалізація стеганографічного алгоритму для цифрового аудіо сигналу	18.04.22	виконано
4	Тестування програмного продукту	02.05.22	виконано
5	Підготовка пояснювальної записки	16.05.22	виконано
6	Підготовка презентації та доповіді	30.05.22	виконано
7	Попередній захист	03.06.22	виконано
8	Нормоконтроль, рецензування	13.06.22	виконано

Здобувач вищої освіти _____ Чураков І.М.

Керівник роботи _____ Трифонова К.О.

ЗАВДАННЯ

на розробку розділу «Охорона праці»

Чуракову Івану Миколайовичу, група РЗ181

Інститут інформаційної безпеки, радіоелектроніки та телекомунікацій

Кафедра кібербезпеки та програмного забезпечення

Тема роботи: *Розробка стеганографічного алгоритму приховування даних в аудіо сигналах*

Зміст розділу:

1. Аналіз умов праці і вибір основних заходів виробничої безпеки.
2. Аналіз пожежної безпеки та вибір заходів і засобів пожежної безпеки.

Керівник роботи	Консультант з охорони праці
_____ _____	_____ _____
(підпис) (прізвище та ініціали)	**(підпис) (прізвище та ініціали)**
« __ » _____20 __ р.	« __ » _____20 __ р.

АНОТАЦІЯ

Кваліфікаційна робота на тему «Розробка стеганографічного алгоритму приховування даних в аудіо сигналах» на здобуття першого (бакалаврського) рівня вищої освіти за спеціальністю 125 – Кібербезпека, спеціалізація, освітня програма: Кібербезпека, містить: 18 рисунків, 1 таблицю, 1 додаток, 23 літературних джерела за переліком посилань. Робота виконана на 62 сторінках загального тексту і 41 сторінках основного тексту.

Метою даної роботи є модифікація стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці цифрового контейнера, для цифрового аудіо сигналу.

Теоретичні методи, які були застосовані в бакалаврській роботі, ґрунтуються на методах цифрової обробки аудіо сигналів, та матричному аналізі. Експериментальне дослідження виконано із застосуванням стороннього програмного забезпечення. Реалізація модифікованого стеганографічного алгоритму, у вигляді програмного застосування із інтерфейсом користувача, виконано за допомогою середовища розробки високого рівня Matlab.

Результатом бакалаврської роботи є програмне застосування для прихованої передачі секретних даних за допомогою цифрових аудіо сигналів, для підвищення захисту інформації у відкритих каналах зв’язку, на основі модифікації стеганографічного методу, заснованого на сингулярному розкладі блоків матриці контейнера.

ЦИФРОВИЙ АУДІО СИГНАЛ, СИНГУЛЯРНИЙ РОЗКЛАД, СТЕГАНОГРАФІЯ, СТЕГАНОСИСТЕМА, ЗАХИСТ ІНФОРМАЦІЇ

ABSTRACT

Qualification work «Development of steganography algorithm for hiding data in audio signals» for the first level of higher education (bachelor) in the specialty 125 – Cybersecurity, specialization, educational program: Cybersecurity, contains: 18 figures, 1 table, 1 appendix, 23 references according to the list of references. Work carried out on 62 total pages of text and 41 pages of main text.

The aim of this work is to modify the steganography algorithm based on the singular decomposition of digital container matrix blocks for digital audio signal.

Theoretical methods used in the bachelor's thesis are based on the methods of digital processing of audio signals and matrix analysis. The experimental study was performed using third-party software. The implementation of the modified steganography algorithm, in the form of a software application with a user interface, is performed using the high-level development environment Matlab.

The result of the bachelor's thesis is a software application for covert transmission of secret data using digital audio signals, to increase the protection of information in open communication channels, based on a modification of the steganography method based on singular decomposition of container matrix blocks.

DIGITAL AUDIO SIGNAL, SINGULAR DECOMPOSITION, STEGANOGRAPHY, STEGANOSYSTEM, INFORMATION PROTECTION

ЗМІСТ

ВСТУП

В мережі Інтернет передається велика кількість файлів. Вони містять інформацію різного виду та можуть стосуватись найрізноманітніших галузей. Значна частина таких файлів представляє собою мультимедійну інформацію, в тому числі різноманітні аудіо файли. У зв'язку, з таким широким розповсюдженням, передачі цифрових аудіо сигналів, актуальним становиться розробка аудіо стеганографічних систем.

Отже, задача прихованої передачі секретних даних, за допомогою цифрових аудіо сигналів, для підвищення захисту інформації у відкритих каналах зв'язку, стає надзвичайно важливою, визначаючи актуальність даної бакалаврської роботи.

Дослідження сучасних стеганографічних алгоритмів для цифрового аудіо сигналу демонструє, що можна виділити дві основні області їх функціонування: область даних та службову область аудіо файлу.

Занурення інформації в службові частини файлу, мають значну перевагу в об'ємі інформації, що буде вбудована при відсутності явних змін даних аудіо файлу. Це означає, що при прослуховуванні такого файлу людина не помітить факт зміни файлу. Натомість програма, що перевірятиме значення полів у відповідності із значеннями формату, без проблем помітить втручання.

Натомість методи, що пов'язані із областями даних, досить чутливі до певних видів атак чи змін, таких як стиснення.

Досліджені недоліки спонукають до розробки нового методу, що буде стійкий до атак чи змін самого файлу.

Тому, метою даної роботи є модифікація стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці цифрового контейнера, для цифрового аудіо сигналу.

Для досягнення поставленої мети необхідно розв'язати наступні задачі:

а) провести аналіз предметної області – проаналізувати існуючи стеганографічні методи для цифрового аудіо сигналу;

б) дослідити реалізацію стеганографічного методу, заснованого на сингулярному розкладі блоків матриці цифрового контейнера;

в) розробити конвертацію вектору значень цифрового аудіо сигналу в матрицю;

г) спроектувати та реалізувати програмне забезпечення, для прихованої передачі інформації за допомогою цифрового аудіо сигналу.

Об'єктом дослідження виступає процедура занурення та вилучення додаткової інформації в цифровим аудіо сигналі.

Предметом дослідження в бакалаврській роботі виступає процес конвертації вектору значень цифрового аудіо сигналу в матрицю.

Теоретичні методи, які були застосовані в бакалаврській роботі, ґрунтуються на методах цифрової обробки аудіо сигналів, та матричному аналізі. Експериментальне дослідження виконано із застосуванням стороннього програмного забезпечення. Реалізація модифікованого стеганографічного алгоритму, у вигляді програмного застосування із інтерфейсом користувача, виконано за допомогою середовища розробки високого рівня Matlab.

Результатом бакалаврської роботи є програмне застосування для прихованої передачі секретних даних за допомогою цифрових аудіо сигналів, для підвищення захисту інформації у відкритих каналах зв’язку, на основі модифікації стеганографічного методу, заснованого на сингулярному розкладі блоків матриці контейнера.

Робота містить наступні основні складові: чотири розділи, вступ та висновок, перелік посилань та додаток.

У вступі визначена мета роботи та основні задачі, реалізація яких призводить до її досягнення, описаний результат роботи.

У першому розділі виконано огляд стеганографії, що працює з цифровими аудіо даними. Розглянуто області застосування стеганосистем. Сформована класифікація стеганографічних методів для занурення повідомлення в аудіо сигнали, на основі дослідженої літератури. Детально досліджено методи вбудовування повідомлення в області даних аудіо файлу та методи вбудовування

секретного повідомлення в службові області аудіо файлу.

У другому розділі представлені теоретичні основи модифікації стеганографічного алгоритму для цифрового аудіо сигналу. Детально досліджено стеганографічний алгоритм, заснований на сингулярному розкладі блоків матриці контейнера, що представляє основу для стеганографічного алгоритму, для цифрового аудіо сигналу. Представлені основні кроки модифікації стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці контейнера. Виконано оцінку ефективності розробленого стеганографічного алгоритму.

У третьому розділі описана практична розробка модифікованого стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці, для цифрового аудіо сигналу. Обґрунтовано вибір середовища розробки, за допомогою якого було створено програмний продукт «Audio Steganography». Представлена організація функціонування програмного забезпечення. Для графічного відображення етапів роботи програми та демонстрації систематичної послідовності виконання поставлених задач, побудовані блок-схема етапу занурення секретного повідомлення та блок-схема вилучення секретного повідомлення. Наведена інструкція користувача програмного забезпечення.

У четвертому розділі розглянуто основні небезпечні фактори при роботі в офісному приміщенні. В результаті аналізу розглянутих факторів, були надані основні рекомендації щодо їх усунення. Особлива увага приділена причинам виникнення та можливим варіантам усунення ризиків виникнення пожежі.

Результати отримані в процесі виконання поданої роботи, представлені в другій частині LXXXV випуску міжнародного наукового журналу «iScience».

1 ПРИХОВУВАННЯ ІНФОРМАЦІЇ В АУДІО ФАЙЛАХ

1.1 Функціонування стеганографічної системи

З розвитком людства, майже всі сфери потребують інформаційного забезпечення. Враховуючи тотальну комп’ютеризацію і використання мережі Інтернет, питання захисту інформації стає все більш актуальним [1].

Щоб забезпечити інформаційну безпеку, розробляються різні методи для її захисту. Їх можна поділити на 2 групи:

– організаційні;

– технічні.

Під організаційними методами захисту інформації, мається на увазі, обмеження доступу до певних ресурсів за допомогою організаційних заходів. Дані методи допоможуть усунути втручання до системи із ненадійних джерел.

Технічні методи реалізують захищеність інформації шляхом використання апаратних та програмних засобів. Серед програмних засобів слід виділити стеганографічні та криптографічні методи.

Стеганографічні методи – це методи, що виконують певні перетворення, в ході яких оригінальний зміст інформації не втрачається. Це дозволяє обмінюватись даними в різних відкритих мережах без ризику, що повідомлення буде знайдене.

Криптографічні методи – це методи, в результаті роботи яких виконується перетворення інформації з використанням спеціальних даних (ключів) для приховування змісту інформації.

На відміну від стеганографії, криптографія не передбачає приховування самої наявності повідомлення [2].

Стеганографічна система або, скорочено, стеганосистема – це організація різноманітних методів та засобів, що застосовуються для побудови системи, яка представляє собою прихований канал передачі інформації [3].

При побудові стеганосистеми мають враховуватися такі положення [4]:

- при розробці системи потрібно врахувати складність математичної реалізації (логічні або алгебраїчні операції за допомогою яких повідомлення буде занурене і вилучено);
- навіть за виявлення фактору наявності повідомлення в контейнері, порушник не повинен мати змогу вилучити повідомлення;
- стеганографічні методи мають забезпечити цілісність даних при передачі контейнера;
- стеганографічний контейнер та канал зв’язку повинні мати прийнятний рівень пропускної здатності;
- при розробці системи варто враховувати модель порушника, якому відомо за наявність приховуваного повідомлення в контейнері, але не відомо в якій саме частині файлу воно знаходиться.

Сьогодні стеганографічні системи активно використовуються для вирішення таких задач захисту інформаційних ресурсів:

- обхід засобів моніторингу;
- захист авторських прав на інтелектуальну власність;
- захист конфіденційної інформації;
- приховування певних програм;
- викрадення інформації (створення невідомих для власника каналів витоку інформації).

Розглянемо структуру стеганографічної системи. Вона передбачає три вхідні дані: ключ, приховувану інформацію та контейнер.

Приховувана інформація – будь-яка закрита інформація, яку необхідно передати і може бути представлена у виді тексту чи файлу. Ключем виступає алгоритм, за допомогою якого будуть занурюватись дані. Під контейнером слід розуміти множину вхідних даних, що буде передаватись по відкритим каналам.

Контейнери бувають двох типів: потокові та фіксовані.

Потокові контейнери – набір даних, який постійно змінюється. Це означає, що на етапі оцінювання неможливо визначити чи поміститься приховувана інформація в контейнер.

Фіксовані контейнери, в свою чергу, мають фіксовану довжину і це означає, що такі контейнери дають змогу вибрати кращий контейнер для повідомлення. Мається на увазі, що при додаванні в контейнер повідомлення, контейнер відчутно не зміниться. Це допоможе надійніше приховати повідомлення. Щоб вибрати кращий контейнер для повідомлення, потрібно оцінити його розмір. Вважається, що якщо контейнер в декілька разів більший ніж повідомлення, то приховуване повідомлення менш помітне.

Типова структура стеганографічної системи зображена на рисунку 1.1 [5].

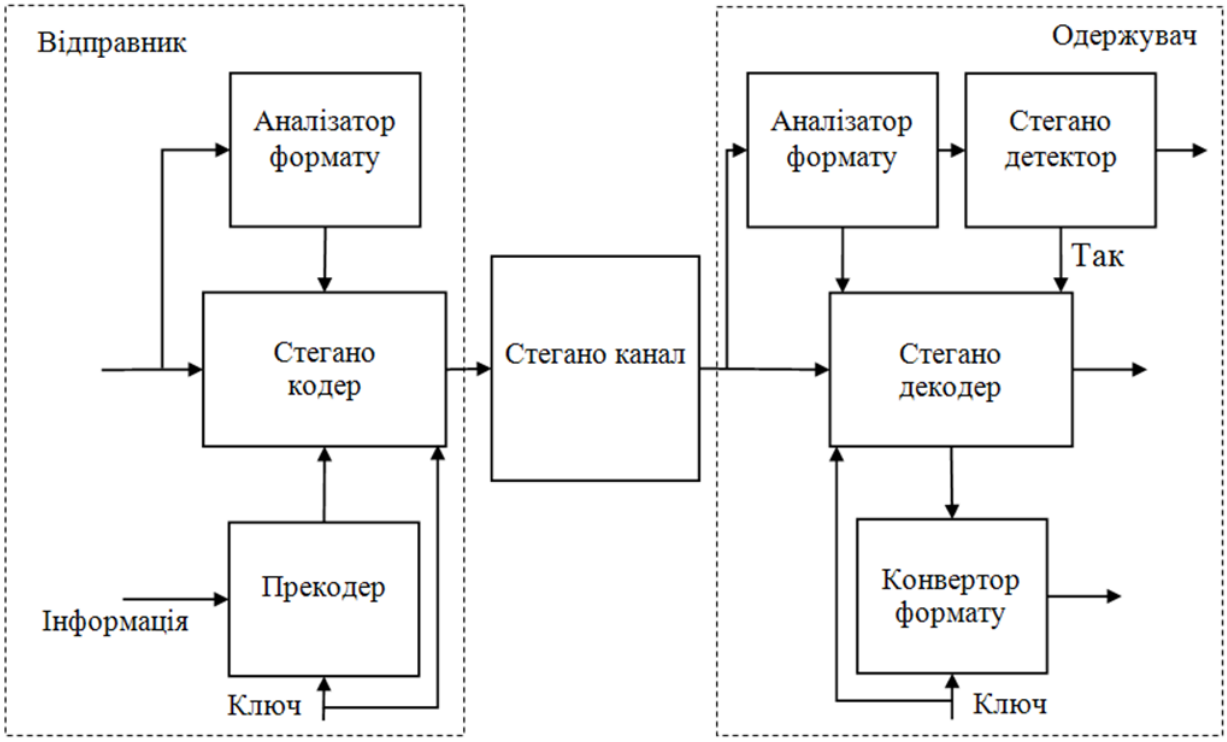


Рисунок 1.1 – Структурна схема стеганографічної системи

Спочатку ключ та інформація, що буде приховуватись, передаються в прекодер. На цьому етапі інформація попередньо шифрується для збільшення надійності. Всього відомо два найпопулярніших види шифрування: з використанням відкритого і закритого ключа. При використанні відкритого ключа, сам ключ передається відкритим каналом разом із самим повідомленням. При використанні закритого ключа, сам ключ зберігається у відправника і одержувача, але не передається по відкритому каналу зв’язку. Потім зашифрована

інформація разом із порожнім контейнером потрапляють до стеганокодеру. Там інформація додається в контейнер і прямує до каналу зв'язку, через який вона доходить до одержувача. Схема для вилучення додаткової інформації схожа на схему для занурення. Спочатку повідомлення надходить до стеганодекодеру. В ньому інформація дістається з контейнера і разом із ключем направляється до конвертеру формату. Там, інформація за допомогою ключа вилучається і одержувач може ознайомитись із прихованим повідомленням.

1.2 Класифікація стеганографічних методів для аудіо файлів

Для розв'язку задачі приховування секретної інформації в цифровому аудіо файлі, дослідники зі всього світу розроблюють методи, що застосовують різноманітні підходи. Стеганографічні методи для цифрових аудіо файлів, інформація про які доступна з відкритих джерел, можна класифікувати у відповідності до області занурення додаткової інформації (рис. 1.2).

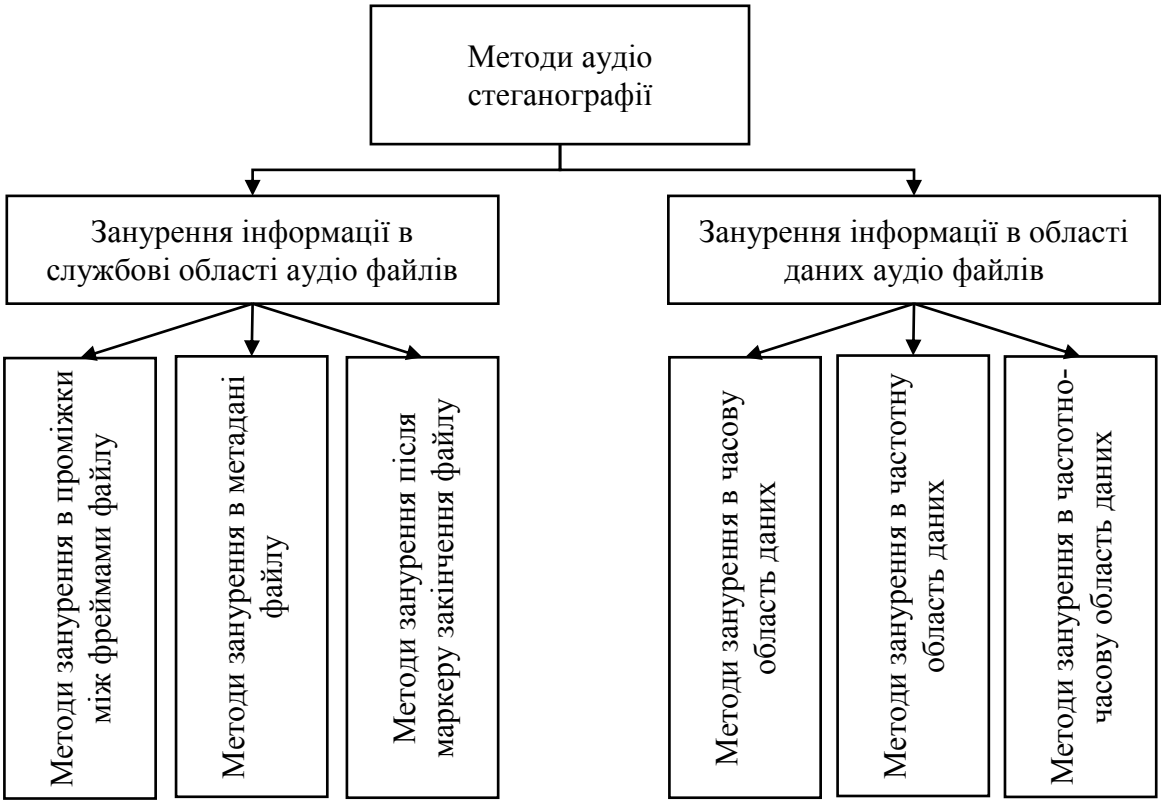


Рисунок 1.2 – Класифікація стеганографічних методів для аудіо файлів

1.3 Стеганографічні методи вбудовування в службові області аудіо файлів

Аудіо файли – файли, які зберігають інформацію про амплітуду і частоту звуку. З цими файлами пов’язана така операція як кодування звуку. Для цього встановлюється певна частота вимірювань і кожен раз записується значення амплітуди у двійковому форматі [6].

Цією операцією займаються спеціальні прилади – аналогово-цифрові перетворювачі (АЦП). Майже всі аудіо файли, що зберігаються на комп’ютері, мають вигляд двійкового коду. Але такий формат не добре підходить для передачі через канали зв’язку, і саме через це, файли піддаються стисненню для подальшого відправлення.

Формат цифрових даних визначається квантуванням, тобто розбиванням значень амплітуди на скінченну кількість інтервалів. Серед усіх видів можна виділити два найбільш популярних: імпульсно-кодова модуляція та сигма-дельта-модуляція. Імпульсно-кодова модуляція – це взяття через однакові проміжки часу відліків рівня аналогового сигналу, з подальшим квантуванням і кодуванням незалежно один від одного [7]. Сигма-дельта модуляція – спосіб модуляції, що забезпечує оцифрування сигналу з заданими характеристиками в робочій смузі частот [7].

Також не варто забувати про надмірність інформації. Вона виникає, коли об’єкт містить занадто багато інформації. Щоб усунути цю проблему використовують аудіокодеки, які надають змогу стиснути сам файл. За стисненням, формати аудіо файлів можна поділити на 3 категорії [7]:

- без стиснення, такі як WAV, AIFF;
- із стисненням без втрат, такі як APE, FLAC;
- із стисненням зі втратами, такі як MP3, OGG.

Службова частина MP3 файлу складається із фреймів, де кожен містить заголовок для зберігання службової інформації та блоку даних. Такий формат збереження даних дозволяє легко помістити приховувану інформацію всередину файлу. Місцями для зберігання приховуваної інформації можуть бути: проміжки

між фреймами, після маркеру закінчення файлу, метадані файлу.

Методи занурення в метадані файлу.

Роздивимось структуру файлу на основі формату MP3. Він містить службову інформацію та користувацьку інформацію. Службова інформація зберігається в певних частинах файлу і називається тегами. Для MP3 формату найбільш популярним форматом метаданих є ID3 (з англ. Identify an MP3). Перша версія цього формату з'явилась ще в 1996. Це було пов'язано з тим, що при появі формату MP3, не було можливості зберігати службову інформацію про файл. Саме щоб вирішити цю проблему Ерік Кемп запропонував додати певну ділянку в пам'яті. Саме так з'явилась перша версія ID3 і мала структуру представлену в таблиці 1.1.

Таблиця 1.1 – Структура формату ID3v1

Поле	Довжина	Опис
заголовок	3	"TAG"
назва	30	30-символьна назва
виконувач	30	30-символьне ім'я виконувача
альбом	30	30-символьна назва альбому
рік	4	строковий запис року
коментар	28 або 30	коментар
нульовий байт	1	якщо номер треку є, цей байт дорівнює 1
track	1	номер треку в альбомі або 0
жанр	1	індекс в списку жанрів або 255

Але дана версія не сильно сподобалась користувачам, оскільки мала ряд недоліків. Перша з них – кількість пам'яті, що виділялась для кожного з атрибутів. Користувачі були вимушені обрізати назви і не тільки. Іншою головною проблемою була відсутність інтернаціональності. Користувачі, як правило, писали на своїй рідній мові і це призводило до пошкодження інформації, що зберігалась [8]. Щоб вирішити дані проблеми, була створена друга версія формату ID3. На відміну від першої версії, друга має динамічні поля, що дозволяють підтримувати потокове відтворювання. Також використання другої версії дозволяє розміщувати службові дані в кінці файлу [9].

Метадані – інформація, що додається до основного файлу, головною метою

якої є опис цього самого файлу. Існує декілька варіантів для приховування інформації.

Перший – додавання інформації в тег ID3v1. Але такий варіант не можна вважати досить надійним. Приховувану інформацію легко знайти навіть через звичайний плеєр. Також потрібно враховувати кількість інформації, яку можна помістити. Тег ID3v1 не може містити багато інформації, що унеможливлює приховування великих обсягів даних.

Другий – додавання інформації в тег ID3v2.4. В порівняні з першим варіантом, другий є більш надійним, оскільки він містить більший об’єм даних. Це означає, що додавання інформації не буде досить помітним [10].

Методи занурення в проміжки між фреймами файлу.

Кожен файл має блочну архітектуру. Ці блоки являють собою фрейми. Фрейми не мають однакового розміру. Але, якщо довжина різна, не вигідно на кожен фрейм виділяти свою кількість пам’яті. Тому, при створенні файлу, є певний сталий розмір кожного фрейму. Якщо фрейм більш короткий, ніж задана довжина, він доповнюється порожнім простором. Також кожен фрейм має заголовки, один з яких (Length) повідомляє, яка саме довжина його фрейму. Програма, яка зчитує ці фрейми, як правило, зчитує довжину вказану в тегу Length і потім ігнорує все що знаходиться до наступного тегу Length. Головна суть метода, вбудувати приховуване повідомлення в проміжки між кінцем фрейму, згідно тегу Length, і початком наступного фрейму [11].

Методи занурення після маркеру закінчення файлу.

Цей метод схожий на минулий за своїм принципом роботи. Єдина різниця полягає в тому, що в якості фрейма виступатиме цілий файл. Деякі формати файлів мають загальний тег Length. Це означає, що при зчитуванні таких файлів програма не буде перевіряти місце, яке було додано. Даний метод можна назвати достатньо надійним, оскільки він не збільшує сильно сам файл, як це робить метод вбудовування інформації між фреймами, тому що інформація не ділиться на частини, а напряму поміщується в кінець файлу. Сам користувач не побачить, з

першого погляду, реальну довжину файлу, оскільки йому буде відображатись довжина, що зберігається в тегу Length.

1.4 Стеганографічні методи вбудовування в області аудіо даних

Методи вбудовування інформації в області даних аудіо файлів набувають все більшої популярності. Це пов’язано з тим, що зрозуміти те що аудіо данні файлу було змінено, не є очевидним. Оскільки плеєри і подібні програми не займаються аналізом даних і не шукають в них стеганоповідомлення. Людське вухо навпаки, не може сприймати широкий діапазон звуку і розпізнати мінімальні відхилення.

Методи занурення в часову область даних.

Одним із перших і найпростіших методів є метод найменшого значущого біта. Суть методу полягає в заміні в кожному байті найменшого значущого біта. Даний варіант достатньо непомітний для людського вуха і має велику пропускну здатність. Але є і певні обмеження. Даний метод неможливо реалізувати для типів файлів, що при збережені будуть проходити через операції стиснення. Такими формати можуть бути MP3, OGG.

Методи занурення в фазову область даних.

Людське вухо не сильно чуттєве до абсолютних значень фаз сигналу. Натомість, воно помічає різницю між цими фрагментами. Виходячи з такої логіки, методи вбудовування в частотну область даних змінюють фази окремих гармонік. Прості варіації цього метода не стійкі до змін. Але більш складні можуть використовувати декілька підходів зміни фаз, що набагато збільшує стійкість самого алгоритму.

Методи занурення в частотно-часову область даних.

Даний метод являє собою комбінацію перших двох методів. Рекомендується вибирати місце занурення, ґрунтуючись на психоакустичній моделі сприйняття звуку. Даний метод є достатньо поширеним і контейнер важко відрізнити від оригіналу, проте є і мінуси. Після обробки втрачається якість звуку, що досить

обмежує вибірку файлів, які можна взяти в якості контейнера.

В даному розділі виконано огляд стеганографії, що працює з цифровими аудіо даними. Розглянуто області застосування стеганосистем. Була сформована класифікація стеганографічних методів для занурення повідомлення в аудіо сигнали, на основі дослідженої літератури. Було детально досліджено методи вбудовування повідомлення в області даних аудіо файлу: часову, частотну, частотно-часову, та методи вбудовування секретного повідомлення в службові області аудіо файлу: мета дані, проміжки між фреймами, кінець файлу.

Занурення інформації в службові частини файлу, мають значну перевагу в об’ємі інформації, що буде вбудована при відсутності явних змін даних аудіо файлу. Це означає, що при прослуховуванні такого файлу людина не помітить факт зміни файлу. Натомість програма, що перевірятиме значення полів у відповідності із значеннями формату, без проблем помітить втручання.

Натомість методи, що пов’язані із областями даних, досить чутливі до певних видів атак чи змін, таких як стиснення.

Досліджені недоліки спонукають до розробки нового методу, що буде стійкий до атак чи змін самого файлу, який продемонстрований в наступному розділі.

2 ТЕОРЕТИЧНІ ОСНОВИ МОДИФІКАЦІЇ СТЕГАНОГРАФІЧНОГО АЛГОРИТМУ ДЛЯ ЦИФРОВОГО АУДІО СИГНАЛУ

2.1 Стеганографічний алгоритм, заснований на сингулярному розкладі блоків матриці контейнера

Звук, як явище, є достатньо природним для людини. Це пов’язано з тим, що людина сприймає звук та аналізує його приблизно все своє життя. Описати звук можна двома шляхами. Інтуїтивне визначення: звук – потік інформації, що отримується людським вухом і в подальшому аналізується мозком. Наукове визначення: звук – це коливання, що поширюються у середовищі.

Для збереження звуку в електронному вигляді, застосовують операції оцифровування звуку. Головна ідея оцифровування полягає в тому, щоб за кожен проміжок часу встановити звуку певне значення, після чого створений вектор значень можна без проблем зберегти. Якщо мова йде про запис звуку, то спочатку з мікрофона отримується значення напруги електричного сигналу протягом всього запису. Дана напруга називається аналоговим представленням звуку. Після цього виконується оцифровування: проводяться заміри напруги сигналу, після цього заміри конвертуються в числа певного діапазону і записуються у файл. Описаний процес називається семплуванням. Дана операція можлива і в зворотному напрямку, коли потрібно програти звук із звукового файлу. В такому випадку конвертуються числові значення семплу (звукового фрагменту) в електричну напругу і безперервно направляються на динаміки.

Людське вухо не здатне чути весь можливий спектр звуку. Людське вухо обмежується значеннями частот від 20 до 22000Гц. Проте чутливість звуку не є сталою у всьому діапазоні значень. Це залежить від зовнішнього середовища, а точніше гучності шуму в ньому. Так, в тихому приміщенні, чутливість вуха максимальна за частот 2-4кГц. Варто також зазначити, що людський голос також не осягає весь можливий частотний діапазон, а може видати значення від 500Гц до 2кГц. Даний факт, що стосується обмеженості людського вуха в плані сприймання звуку певних частотних діапазонів, дає можливість створити

алгоритм для стиснення звуку із втратами. Головна ідея стиснення – відкидання семплів, які людське вухо не може почути.

Стиснення може бути реалізоване із втратами і без них. Останнє залежить від самого аудіо файлу, оскільки не кожен запис можна стиснути без втрат. Від самого запису також залежить алгоритм, яким саме буде проводитись стиснення.

Розглядають наступні види стиснення [12-13]:

- кодування повторів – його краще за все використовувати, коли запис містить звуки, що повторюються багато разів;
- статистичні методи – кожному звуковому семплу на всій ділянці запису присвоюється певний код; його значення та різноманітність (тобто діапазон можливих значень) залежить від того, скільки біт виділено на один семпл;
- словниковий підхід – даний алгоритм ґрунтується на припущені, що певні звуки будуть повторюватись часто протягом всього запису.

Останнім часом одним з найпоширеніших стандартів стиску для цифрового аудіо звуку став MPEG. В основі даного стиснення лежить принцип квантування. Проте принцип квантування працює зі значеннями звукових частот, а не із семплами. Ці частоти вираховуються із семплів звукового файлу. Виходячи з того факту, що кодеру наперед відомо коефіцієнт стиснення, кодер в кожен період часу знає скільки біт можливо виділити квантованому сигналу. Важливою частиною кодера є алгоритм призначення бітів. Даний алгоритм використовує частотний спектр останнього семплу для того, щоб визначення розміру квантованого сигналу, щоб шум, який з’явиться в результаті, був не чутний для людини.

Схожий принцип квантування використовується і для стиснення зображень. Також атаки стисненням, що для звуку, так і для зображення, є найбільш популярними для стеганографії. Проте атаки стисненням у випадку зображення є більш дослідженими, на відміну від звуку.

Так, в роботі [14] було запропоновано стеганографічний алгоритм для зображення, який є не чутливий до стиснення в певному діапазоні. В його основі

лежить сингулярне розкладання матриці. Сингулярним розкладанням матриці називають представлення матриці A, у вигляді:

$$A=U\Sigma V^T, \tag{2.1}$$

де U – матриця розміром $m \times n$, яка задовольняє відношенню $U^TU=I$;

V – квадратна матриця порядку n, яка задовольняє відношенню $V^TV=I$,

Σ – діагональна матриця, що містить сингулярні числа.

Основні кроки алгоритму для занурення секретного повідомлення наступні:

а) розбити початкову матрицю певної кольорової компоненти зображення на блоки f розміром 8×8, що не будуть перетинатись. Кожен блок буде зберігати один біт повідомлення;

б) конвертувати повідомлення у вектор, що буде містити значення 0 або 1;

в) для кожного блок f побудувати сингулярне розкладання;

г) якщо біт інформації, що потрібно помістити дорівнює нулю, то:

$$\bar{\sigma}_1=roundn(\sigma_1,K)+\frac{1}{4}\cdot\sigma_2, \tag{2.2}$$

де σ_1 – перше сингулярне число;

σ_2 – друге сингулярне число;

$\bar{\sigma}_1$ – перше сингулярне число після зміни;

K – коефіцієнт округлення;

roundn() – функція округлення до розряду K.

д) якщо біт інформації, що потрібно помістити дорівнює одиниці, то:

$$\bar{\sigma}_1=roundn(\sigma_1,K)+\frac{3}{4}\cdot\sigma_2. \tag{2.3}$$

е) надати блоку матриці початкового вигляду використавши формулу:

$$\bar{f}=U\bar{\Sigma}V^T, \tag{2.4}$$

де $\bar{f}$ – блок матриці після занурення;

$\bar{\Sigma}$ – матриця сингулярних чисел після занурення.

ж) зібрати блоки розміром 8×8 в матрицю цифрового зображення.

Алгоритм вилучення секретного повідомлення має наступний вигляд:

а) розбити початкову матрицю певної кольорової компоненти зображення на блоки f розміром 8×8, що не будуть перетинатись;

б) для кожного блок *f* побудувати сингулярне розкладання;

в) якщо виконується умова:

$$\overline{\overline{(\sigma_1}}\text{-}roundn(\overline{\overline{\sigma_1}}, K)) < \frac{1}{2} \cdot \bar{\sigma}_2 , \quad (2.5)$$

тоді отримано 0, інакше – 1;

г) конвертувати вектор, що складається з 0 та 1, в текст.

2.2 Модифікація стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці контейнера

У попередньому розділі виконано детальне дослідження стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці цифрового зображення. Поданий алгоритм було обрано, завдяки його стійкості до атаки стисненням для певного діапазону коефіцієнта стиснення, як встановлено авторами поданого алгоритму. Атаки стисненням є найбільш популярними атаками і для цифрових аудіо сигналів. Але для поданого виду сигналу дана атака є найменш дослідженою. Завдяки загальній ідеї алгоритму стиснення, що базується на квантуванні частотних коефіцієнтів, запропоновано виконати застосування стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці цифрового зображення для іншого виду сигналу, а саме цифрового аудіо сигналу. Для реалізації поданої ідеї, запропоновано виконати модифікацію стеганографічного алгоритму, в результаті розробки алгоритмів конвертування вектору значень цифрового аудіо сигналу в блоки матриці.

Визначено три способи конвертування: діагональний; горизонтальний; вертикальний. Розглянемо кожен з них більш детально.

Діагональне. Для даного способу передбачається розбити вектор значень цифрового аудіо сигналу на вектори довжиною 64 значення. Для кожного отриманого вектору, поставити у відповідність блок матриці розміром 8 на 8. Матриця заповнюється значенням з вектору, розмір якого 64 елемента, послідовно з лівого верхнього кута змійкою до правого нижнього кута. Блок-

схема відповідного алгоритму представлена на рисунку 2.1.

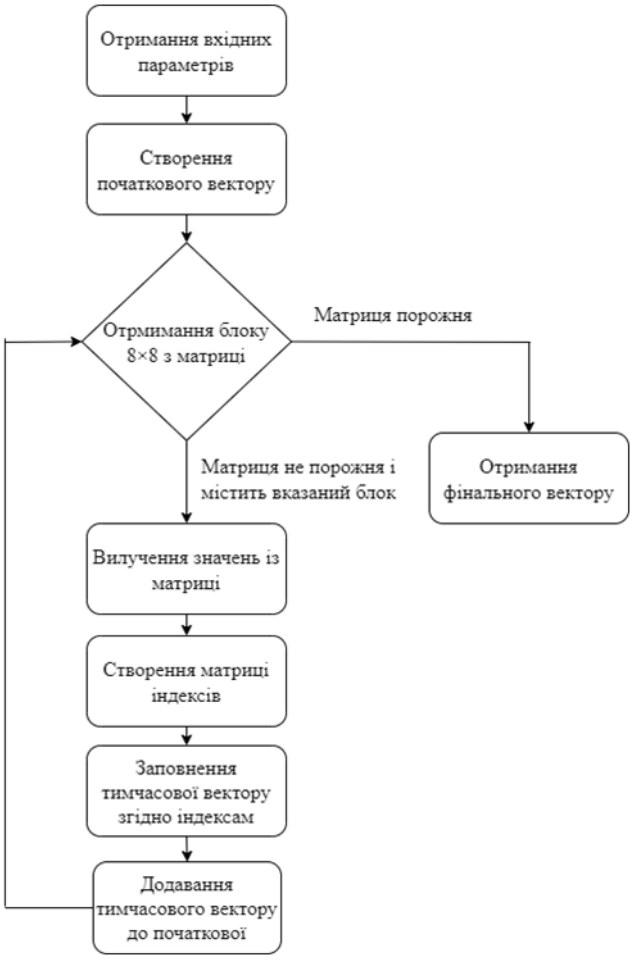
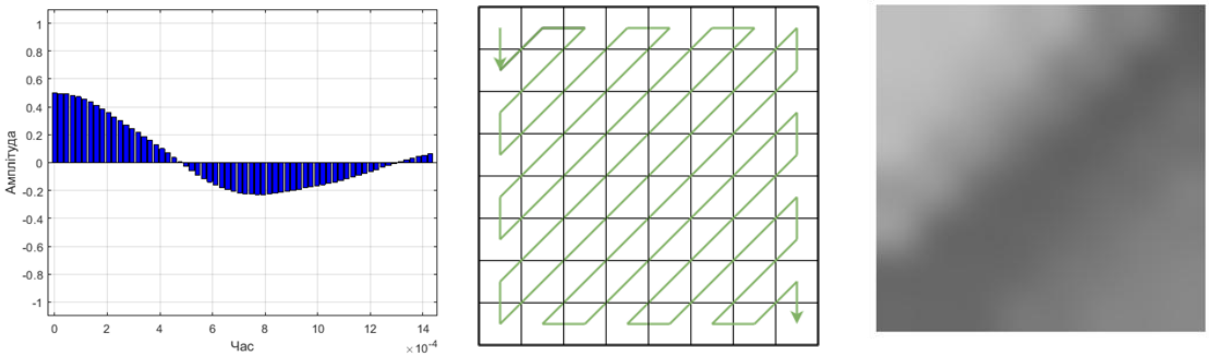


Рисунок 2.1 – Блок-схема алгоритму діагонального конвертування

На рисунку 2.2 наведено приклад діагонального конвертування.



а) б) в)

Рисунок 2.2 – Демонстрація діагонального конвертування а) – цифровий аудіо сигнал; б) – схема конвертування; в) – представлення матриці

Горизонтальний. Для даного способу передбачається розбити вектор значень цифрового аудіо сигналу на вектори довжиною 64 значення. Для кожного отриманого вектору, поставити у відповідність блок матриці розміром 8 на 8. Матриця заповнюється значенням з вектору, розмір якого 64 елемента, послідовно у горизонтальному напрямку. Блок-схема відповідного алгоритму представлена на рисунку 2.3.

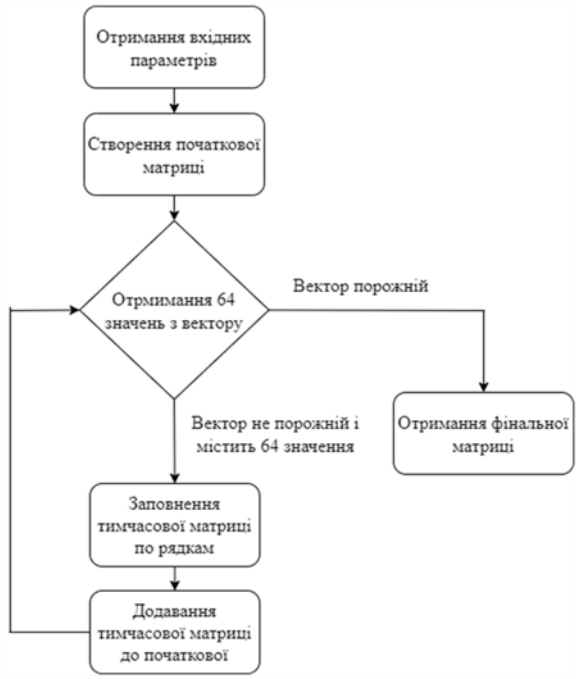
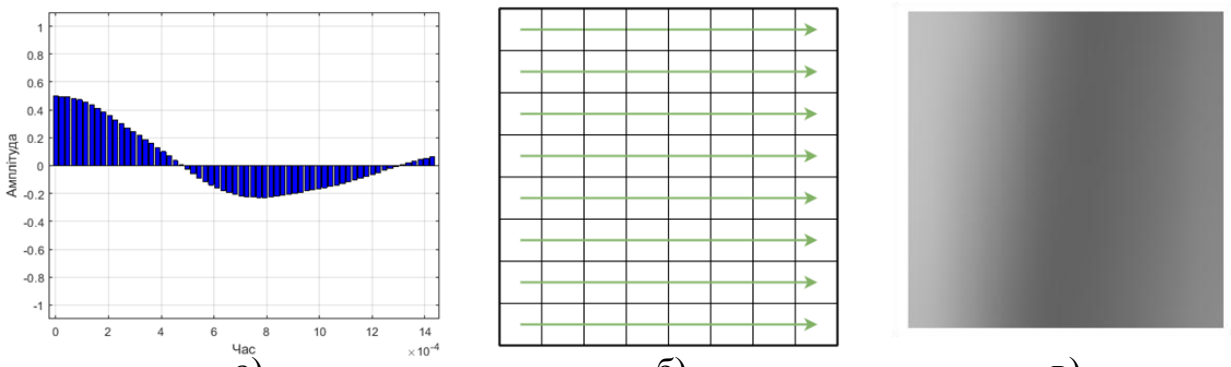


Рисунок 2.3 – Блок-схема алгоритму горизонтального конвертування

На рисунку 2.4 наведено приклад горизонтального конвертування.



а) б) в)

Рисунок 2.4 – Демонстрація горизонтального конвертування а) – цифровий аудіо сигнал; б) – схема конвертування; в) – представлення матриці

Вертикальний. Для даного способу передбачається розбити вектор значень цифрового аудіо сигналу на вектори довжиною 64 значення. Для кожного отриманого вектору, поставити у відповідність блок матриці розміром 8 на 8. Матриця заповнюється значенням з вектору, розмір якого 64 елемента, послідовно у вертикальному напрямку. Блок-схема відповідного алгоритму представлена на рисунку 2.5.

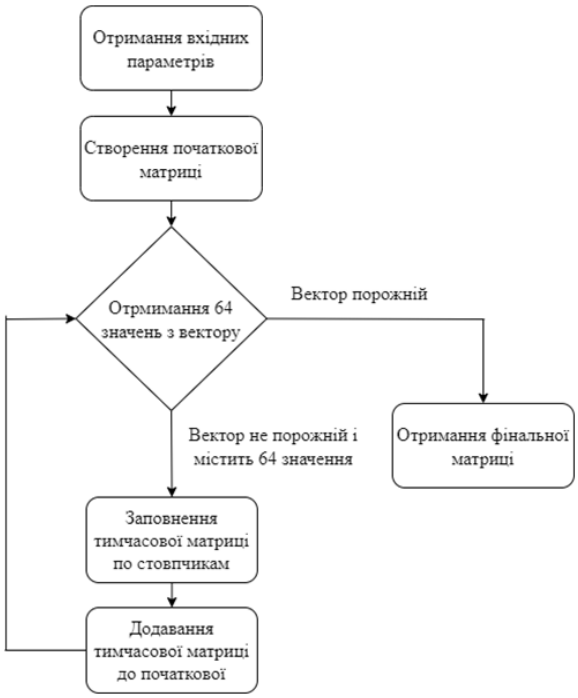
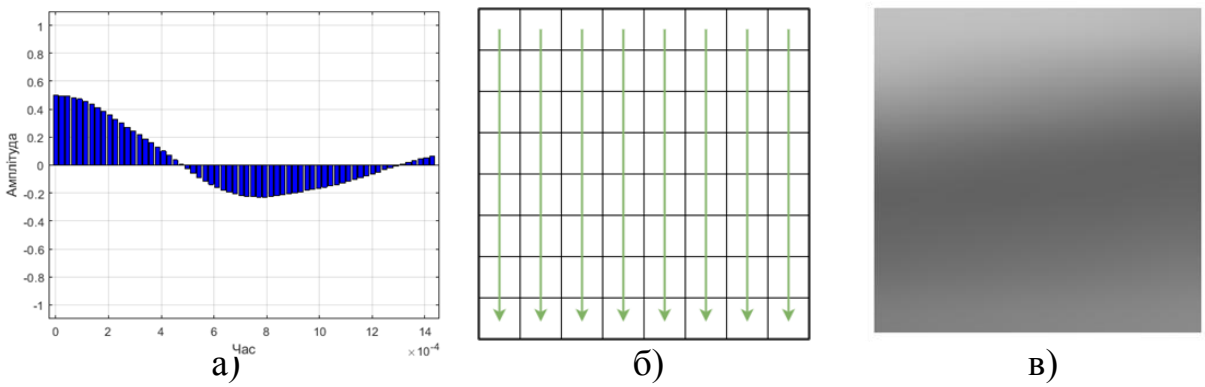


Рисунок 2.5 – Блок-схема алгоритму вертикального конвертування

На рисунку 2.6 наведено приклад вертикального конвертування.



а) б) в)

Рисунок 2.6 – Демонстрація вертикального конвертування а) – цифровий аудіо сигнал; б) – схема конвертування; в) – представлення матриці

Основні кроки модифікованого стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці контейнера наведені нижче.

Занурення секретного повідомлення в цифровий аудіо сигнал наступні:

а) виконати конвертування цифрового аудіо сигналу, у відповідності до обраного способу, в блоки матриці f розміром 8×8. Кожен блок буде зберігати один біт повідомлення;

б) конвертувати секретне повідомлення у вектор, що буде містити значення 0 або 1;

в) для кожного блок f побудувати сингулярне розкладання;

г) якщо біт інформації, що потрібно помістити дорівнює нулю, то:

$$\bar{\sigma}_1 = roundn(\sigma_1, K) + \frac{1}{4} \cdot \sigma_2, \tag{2.6}$$

де σ_1 – перше сингулярне число;

σ_2 – друге сингулярне число;

$\bar{\sigma}_1$ – перше сингулярне число після зміни;

K – коефіцієнт округлення;

$roundn()$ – функція округлення до розряду K.

д) якщо біт інформації, що потрібно помістити дорівнює одиниці, то:

$$\bar{\sigma}_1 = roundn(\sigma_1, K) + \frac{3}{4} \cdot \sigma_2. \tag{2.7}$$

е) надати блоку матриці початкового вигляду використавши формулу:

$$\bar{f} = U\bar{\Sigma}V^T, \tag{2.8}$$

де $\bar{f}$ – блок матриці після занурення;

$\bar{\Sigma}$ – матриця сингулярних чисел після занурення.

ж) виконати обернене конвертування, у відповідності до обраного способу, з блоків матриць $\bar{f}$ розміром 8×8, у вектори розміром 64;

з) зібрати окремі вектори розміром 64, в вихідний цифровий аудіо сигнал.

Алгоритм вилучення секретного повідомлення з цифрового аудіо сигналу буде мати наступний вигляд:

а) виконати конвертування цифрового аудіо сигналу, у відповідності до обраного способу, в блоки матриці f розміром 8×8;

б) для кожного блок *f* побудувати сингулярне розкладання;

в) якщо виконується умова:

$$\overline{\overline{(\sigma_1}}\text{-}roundn(\bar{\bar{\sigma}}_1, K)) < \frac{1}{2} \cdot \bar{\sigma}_2 , \qquad (2.9)$$

тоді отримано 0, інакше – 1;

г) конвертувати вектор, що складається з 0 та 1, назад до початкового тексту.

2.3 Ефективність розробленого стеганографічного алгоритму

Для порівняння роботи модифікованого стеганографічного алгоритму А0, заснованого на сингулярному розкладі блоків матриці цифрового контейнера, з стеганографічними програмними застосуваннями, доступними в мережі Інтернет [15-18], обрані наступні:

- А1 – DeepSound [17];
- А2 – OpenPuff [18].

DeepSound – проста в застосуванні програма для занурення та вилучення секретної інформації для аудіо файлів. Додатково, DeepSound надає можливість конвертувати аудіо файл. З мінусів можна зазначити, що немає можливості вказувати папку для вихідних файлів. Програма має ряд переваг в порівнянні з іншими програмами, а саме: можливість працювати з різними форматами аудіо файлів (flac, mp3, wma, wav, ape); надає можливість занурювати цілі файли, а не лише їх зміст, у вигляді, наприклад тексту. Для більшої надійності, програма може додатково зашифрувати повідомлення, використовуючи симетричний алгоритм блочного шифрування AES-256.

OpenPuff – безкоштовне стеганографічне програмне застосування, розроблене Cosimo Oliboni, для приховування інформації в аудіо файлах. Програма дозволяє виконувати приховування секретної інформації в більше ніж одному файлі. Остання версія програми дозволяє працювати з наступними типами файлів: flac, mp3, wav.

Для визначення ефективності розглянутих алгоритмів, з точки зору надійності сприйняття побудованого стеганографічного повідомлення, застосовано показник пікового відношення «сигнал-шум» PSNR.

Проведено обчислювальний експеримент, в якому було використано сто цифрових аудіо файлів. За допомогою трьох, вказаних вище, стеганографічних програм було виконано занурення ста різних текстових повідомлень. Отримані стеганоповідомлення були збережені у вигляді цифрових аудіо файлів. За допомогою додаткового сервісу www.compresss.com, отримані стеганоповідомлення пройшли процедуру стиснення, з різними коефіцієнтами. Для отриманих аудіо сигналів виконано розрахунок PSNR. Встановлено усереднене значення коефіцієнту PSNR по всім тестовим аудіо сигналам для кожного окремого програмного забезпечення, в умовах заданого коефіцієнту стиснення. Результати експерименту представлені на рисунку 2.7.

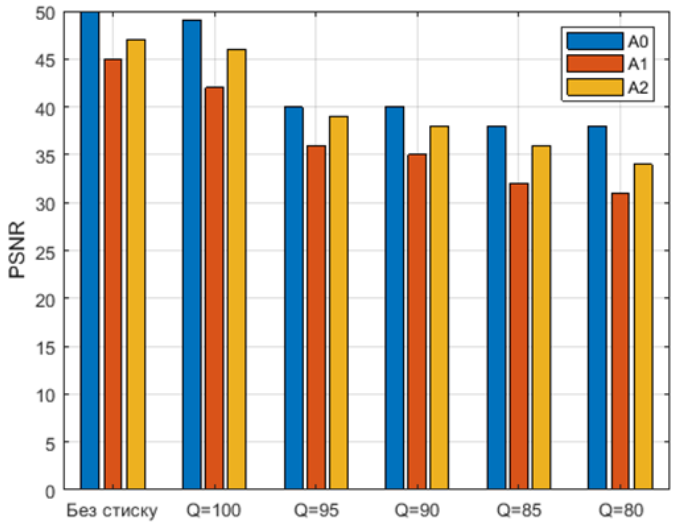


Рисунок 2.7 – Ефективність стеганографічних програмних застосувань для цифрового аудіо сигналу

У відкритому доступі, в мережі Інтернет, вдалося знайти лише дві програми DeepSound та OpenPuff, для виконання стеганографічного занурення та вилучення секретного повідомлення в цифрові аудіо файли, що дозволяють працювати з файлами в форматі з втратами. Жодної інформації про роботу алгоритмів, що

реалізовані в програмах DeepSound та OpenPuff, виявити не вдалося. Тобто, для поданих програм визначена можливість роботи з файлами у форматі з втратами, але немає інформації про стійкість результатів роботи програм до атак стиском.

В розділі продемонстровані теоретичні основи модифікації стеганографічного алгоритму для цифрового аудіо сигналу.

Виконано детальне дослідження стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці цифрового зображення. Головною особливістю якого є стійкість до атак стиском, для деякого діапазону коефіцієнту стиску.

Представлені основні кроки модифікації стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці контейнера. Виконано оцінку ефективності розробленого стеганографічного алгоритму в порівнянні з іншими стеганографічними програмними застосуваннями.

3 ПРАКТИЧНА РОЗРОБКА МОДИФІКОВАНОГО СТЕГАНОГРАФІЧНОГО АЛГОРИТМУ ДЛЯ ЦИФРОВОГО АУДІО СИГНАЛУ

3.1 Середовище розробки програмного забезпечення

Для реалізації модифікованого стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці, для цифрового аудіо сигналу, було обрано середовище розробки Matlab.

Matlab – це платформа для розробки програмного продукту, яка призначена для вирішення інженерних або математичних задач [19].

Мова. Дана мова програмування, через свою високорівневість, дозволяє на пряму контролювати потоки, дані та функції враховуючи об’єктно-орієнтоване програмування. Завдяки такому функціоналу програміст, що використовує Matlab, не обмежений в масштабах своєї програми. Він може створювати, як прості консольні застосування, так і великі застосування, з використанням графічного інтерфейсу.

Середа розробки. Це набір інструментів, що використовуються для надання усіх можливостей, як мови програмування, так і вбудованих бібліотек. Використовуючи дану середу розробки, розробник може відстежувати змінні під час виконання програмного коду, переглядати роботу застосування, використовуючи режим відладки.

Керована графіка. Використовуючи Matlab, можна обробляти зображення, створювати дво- або тривимірні графіки, та реалізовувати для них анімацію. Але на цьому функціонал не закінчується. Matlab надає інструменти для тонкого налаштування зовнішнього вигляду цих самих графіків та для побудови користувацького графічного інтерфейсу.

Бібліотека функцій математики. Це збірник, що містить програмну реалізацію майже всіх математичних функцій від найпростіших, таких як добуток або різниця, до більш складних, таких як сингулярне розкладання матриці або її транспонування.

Інтерфейс програми. Дана бібліотека дозволяє створювати програми на таких мовах програмування як C та Fortran. При цьому використовуючи Matlab як обчислювальний інструмент.

Для спрощення взаємодії із розробленим стеганографічним алгоритмом занурення секретної інформації в цифровий аудіо сигнал, було вирішено створити користувацький графічний інтерфейс. Matlab має вбудований конструктор для побудови інтерфейсу, що називається Guide.

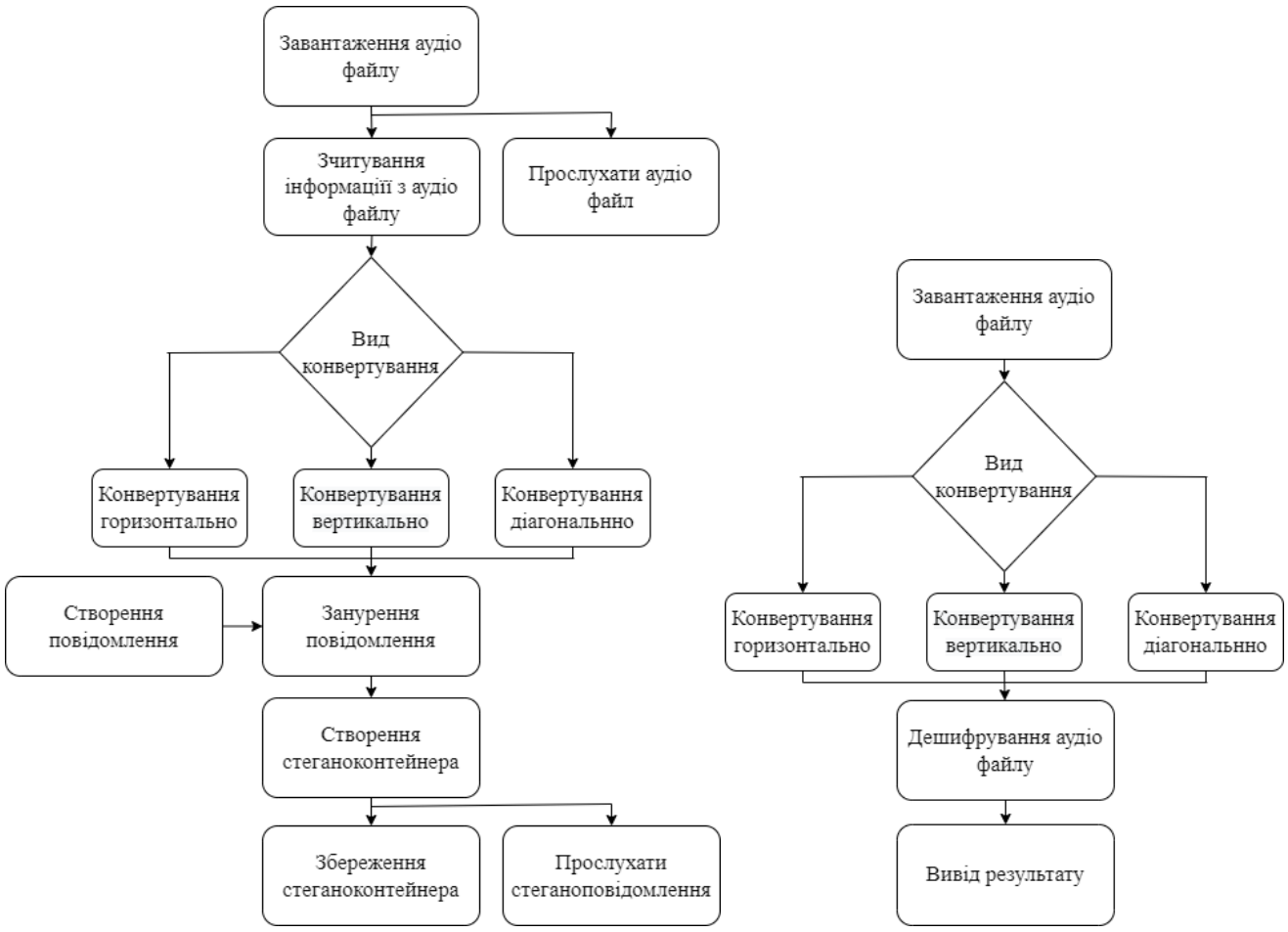
Guide. Це зручний конструктор для візуального програмування. Елементи, що мають бути на формі, можна просто перенести на саму форму. Також він надає зручну можливість оброблювати різного роду події. При використанні Guide, програміст не обмежений створенням застосування, що буде містити одну форму. Тобто є можливість створювати застосування, що складається з декількох вікон. Також даний конструктор дозволяє використовувати стандартні діалогові вікна, такі як підтвердження або вибір файлу [20].

Розробка програмного застосування, що реалізує модифікований стеганографічний алгоритм, напряму пов’язана з взаємодією із звуковими сигналами. Matlab має вбудований спектр можливостей для вирішення різноманітних задач, що пов'язані зі звуковими файлами.

Робота із аудіо. Функціонал, що надає Matlab, для роботи із звуком достатньо широкий. Завдяки функції sound() та її перевантаженням можливо відтворювати звук вказавши при цьому додаткові параметри, такі як розрядність, частоту дискретизації, а в нових версіях, ще й можливе масштабування. Завдяки функції wavread можливо зчитувати амплітуди звукового файлу з розширенням wav. Даний метод, так само, як і попередній, має перевантаження, які дозволяють отримати із звукового файлу дискретизований звук, розрядність та структуру. Але на цьому функціонал Matlab не закінчується. Крім відтворення та зчитування є можливість створення аудіо файлу із вектору завдяки функції wavwrite, яка завдяки своїм перевантаженням дозволяє додатково вказувати дискретизацію та розрядність [21].

3.2 Організація функціонування програмного забезпечення

У відповідності до типового циклу розробки програмного забезпечення, перед кроком розробки, виконують проектування результуючого програмного застосування. Отже, на стадії проектування, при побудові програмного застосування, що реалізує стеганографічний алгоритм для аудіо сигналів, виконано виділення двох основних етапів роботи програми. Перший – виконання занурення секретної інформації до цифрового аудіо сигналу. Другий – вилучення секретного повідомлення з цифрового аудіо сигналу. Для графічного відображення поданих етапів та демонстрації систематичної послідовності виконання поставлених задач, побудовані відповідні блок-схеми (рис. 3.1).



а) б)

Рисунок 3.1 – Блок-схема алгоритму роботи стеганографічної програми а) – занурення секретного повідомлення; б) – вилучення секретного повідомлення

Блок-схема реалізації занурення повідомлення, містить наступні процеси:

– завантаження аудіо файлу – зчитування файлу, що надає користувач програми, шляхом вибору його в діалоговому вікні;
– зчитування інформації з аудіо файлу – отримання інформації про файл, такої, як його амплітуда, частота, канали звукового сигналу;
– прослуховування аудіо файлу – програвання аудіо файлу, що не містить приховуваного повідомлення;
– конвертування горизонтально, вертикально, діагонально – до вектору значень амплітуди аудіо сигналу використовується одне з трьох видів конвертування, в залежності від вибору користувача;
– створення повідомлення – ввід тексту використовуючи латинський алфавіт, що буде занурено в завантажений аудіо файл;
– занурення повідомлення – у відповідності до розробленого алгоритму виконується занурення секретного повідомлення;
– створення аудіоконтейнера – конвертування блоків назад у вектор і створення нового аудіо файлу;
– збереження стеганоконтейнера – збереження нового аудіо файлу, що містить секретне повідомлення, за адресою, вказаною користувачем;
– прослуховування стеганоповідомлення – програвання аудіо файлу, що містить приховане повідомлення.

Блок-схема реалізації вилучення повідомлення, містить наступні процеси:

– завантаження аудіо файлу – зчитування файлу, що містить приховане повідомлення, та отримання вектору значень амплітуди;
– конвертування горизонтально, вертикально, діагонально – вибір виду конвертування, що було застосовано при зануренні повідомлення;
– дешифрування аудіо файлу – у відповідності до розробленого алгоритму, вилучення повідомлення та конвертування до латинського алфавіту;
– вивід результату – вивід повідомлення, що було приховано, у спеціальному вікні.

3.3 Інструкція користувача програмного забезпечення

Для практичного застосування та дослідження стеганографічного алгоритму для цифрових аудіо сигналів, що представляє собою модифікацію стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці цифрового зображення, розроблено програмний продукт «Audio Steganography».

Інтерфейс програмного застосування зображено на рисунку 3.2, програмний код знаходиться в додатку А.

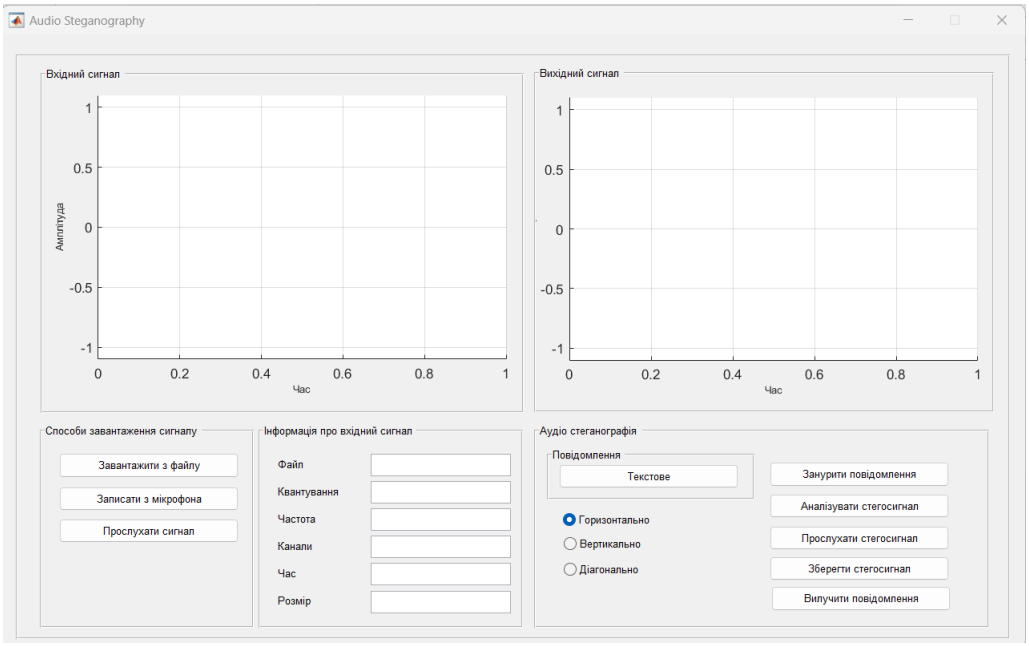


Рисунок 3.2 – Інтерфейс програмного застосування «Audio Steganography»

Для завантаження цифрового аудіо файлу, необхідно натиснути на кнопку «Завантажити з файлу», після чого в діалоговому вікні вибрати потрібний файл. В результаті виконаної операції, в області, що позначена як «Вхідний сигнал», буде побудовано графік значень аудіо сигналу. Ось абсцис відповідає за значення часу. Ось ординат – за значення амплітуди. В області «Інформація про вхідний сигнал» надається коротка інформація про завантажений аудіо файл. Програмне застосування для подальшої обробки дозволяє виконувати завантаження аудіо файлів різного формату (рис. 3.3).

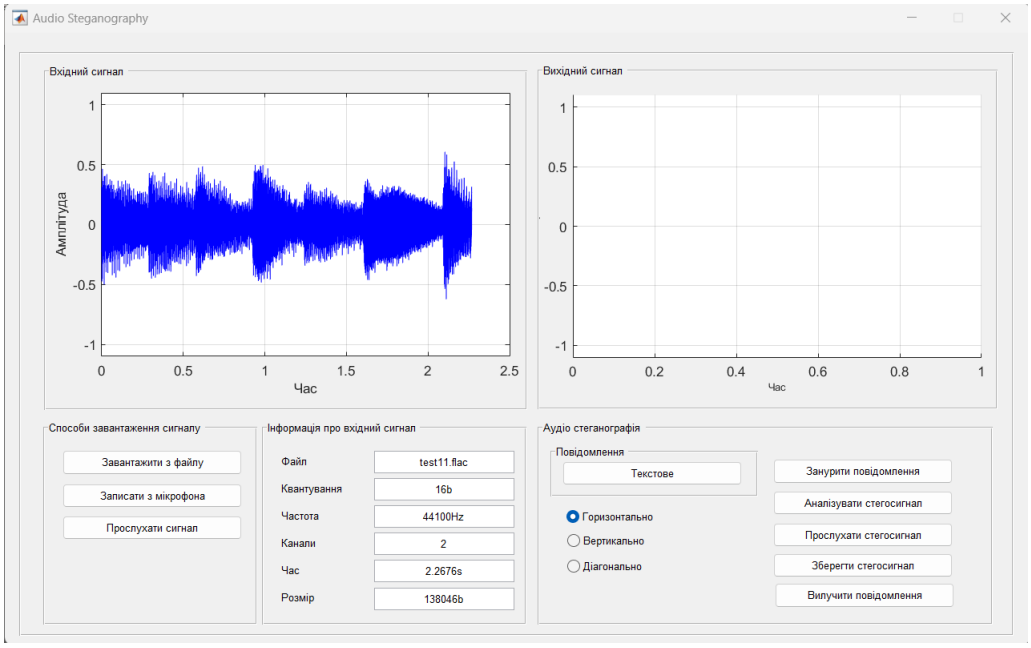


Рисунок 3.3 – Інтерфейс програмного застосування завантаження аудіо файлу

Реалізовано додаткову можливість завантаження цифрового аудіо файлу, що представляє собою запис голосу людини через мікрофон. Для цього потрібно натиснути на кнопку «Записати з мікрофона». Після натиснення з’явиться діалогове вікно (рис 3.4), яке надає простий функціонал для запису та збереження щойно створеного цифрового аудіо файлу.

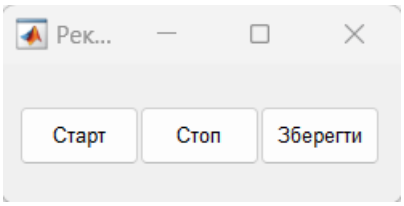


Рисунок 3.4 – Інтерфейс діалогового вікна для запису з мікрофона

Після завантаження файлу чи запису з мікрофона, надається можливість прослухати аудіо сигнал, натиснувши на кнопку «Прослухати сигнал».

Завантаження секретного повідомлення, що використовується для процедури занурення стеганографічного алгоритму, відбувається в результаті натискання кнопки «Текстове» в розділі «Повідомлення». Подана дія призводить

до відображення діалогового вікна для введення тексту повідомлення (рис. 3.5).

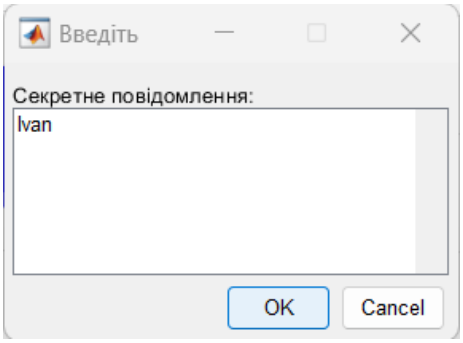


Рисунок 3.5 – Інтерфейс діалогового вікна вводу секретного повідомлення

Для проведення дослідження стеганографічного алгоритму, в програмному застосуванні реалізовано три можливості для конвертування вектору значень аудіо файлу в блоки матриці: горизонтальний, вертикальний, діагональний.

Реалізація процедури занурення секретного повідомлення, відбувається при натисканні кнопки «Занурити повідомлення». В результаті виконаної операції, в області, що позначена як «Вихідний сигнал», буде побудовано графік значень аудіо сигналу, що представляє собою стеганоповідомлення (рис. 3.6).

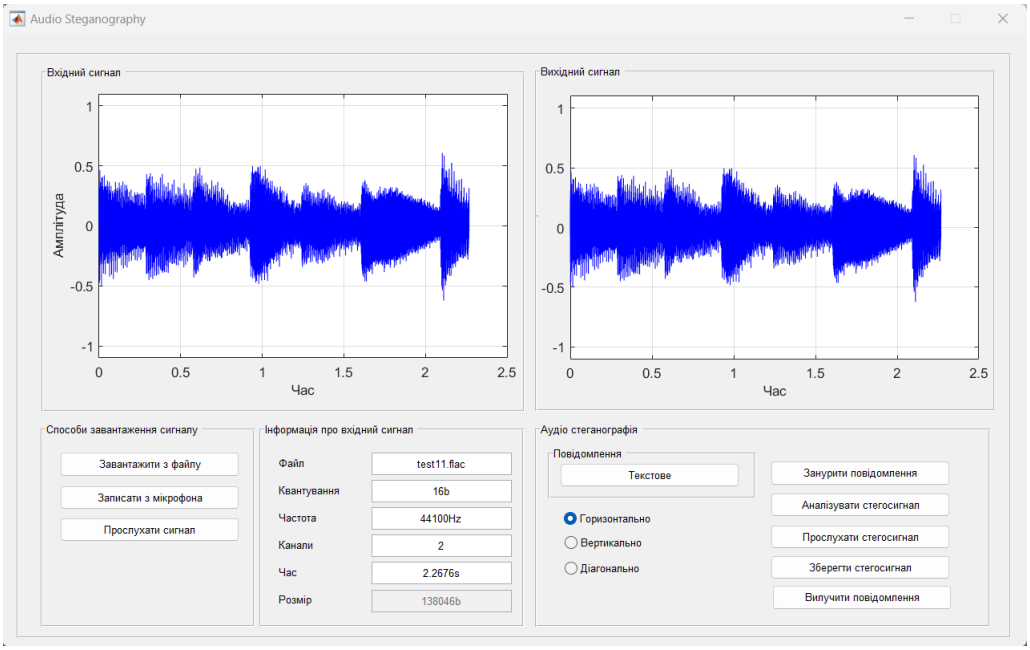


Рисунок 3.6 – Інтерфейс програмного застосування після занурення секретного повідомлення

Основою стеганографічного алгоритму, що використовується для занурення додаткової інформації, є сингулярний розклад блоку матриці, спосіб побудови якого обирається користувачем: горизонтальний, вертикальний, діагональний. Демонстрація набору сингулярних чисел будь-якого блоку, реалізована за допомогою кнопки «Аналізувати стегосигнал».

При натисканні на кнопку, відображається діалогове вікно з можливістю вибору номера блоку (рис. 3.7).

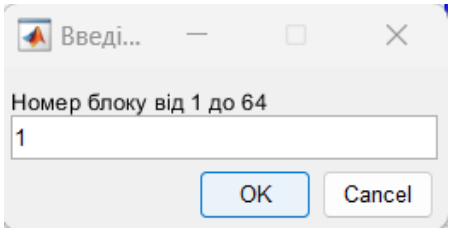


Рисунок 3.7 – Інтерфейс діалогового вікна вибору блоку для аналізу

У відповідності до введеного номеру, обирається блок матриці, для якого виконується сингулярний розклад для встановлення набору сингулярних чисел.

Для демонстрації отриманого результату, виводиться окреме вікно з ілюстрацією блока матриці в градації сірого, отриманого в результаті обраного способу конвертації, та графік значень сингулярних чисел (рис. 3.8).

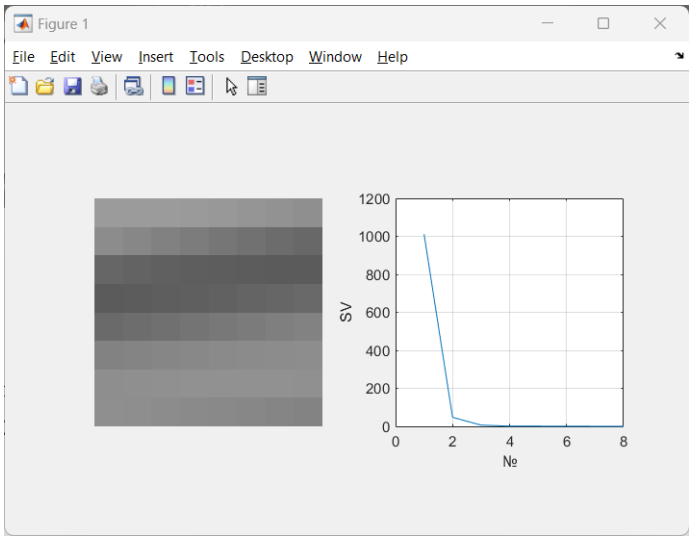


Рисунок 3.8 – Інтерфейс програмного застосування для демонстрації аналізу блока матриці конвертованого цифрового аудіо сигналу

Реалізовано додаткову можливість з прослуховування отриманого стеганоповідомлення, за допомогою кнопки «Прослухати стеганосигнал».

Для подальшого застосування отриманого стеганоповідомлення, реалізована кнопка «Зберегти стегосигнал», для збереження стеганоповідомлення у вигляді цифрового аудіо файлу.

Процедура вилучення секретного повідомлення реалізована за допомогою кнопки «Вилучити повідомлення». В результаті виконання поданої дії, в діалоговому вікні відображається отримане секретне повідомлення (рис 3.9).

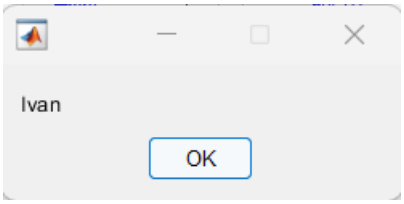


Рисунок 3.9 – Інтерфейс діалогового вікна виводу секретного повідомлення

В даному розділі представлена практична розробка модифікованого стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці, для цифрового аудіо сигналу.

Описано середовище розробки, за допомогою якого було створено даний програмний продукт «Audio Steganography». Обґрунтовано, чому саме Matlab був вибраний, як середа для розробки програми.

Представлена організація функціонування програмного забезпечення. Для графічного відображення етапів роботи програми та демонстрації систематичної послідовності виконання поставлених задач, побудовані блок-схема етапу занурення секретного повідомлення та блок-схема вилучення секретного повідомлення.

Наведена інструкція користувача програмного забезпечення.

4 ОХОРОНА ПРАЦІ

Високий рівень розвитку технологій не може надати стовідсоткових гарантій безпеки для людини на виробництві. Але може дозволити притримуватись стандартів та норм, для того щоб наблизити рівень ризику до прийнятного значення, щоб відповідати європейським та світовим вимогам.

Виходячи з того, що результатом проведеної роботи буде програмний продукт, то робочою зоною виступатиме офісне приміщення. Умови праці для працівників можна вважати наступними: звичайний об’єм приміщення, користування розробленим програмним продуктом буде відбуватись через взаємодію із персональною технікою (або технікою, що може бути видана самою компанією, яка орендує офіс).

Фізичними небезпечними і шкідливими виробничими факторами для працівників в офісних приміщеннях згідно стандарту можуть бути:

- підвищена або понижена температура повітря робочої зони;
- підвищена або понижена вологість повітря;
- підвищене значення напруги в електричній мережі;
- відсутність або нестача природного світла;
- недостатня освітленість робочої зони;

Відсутність або нестача природного світла.

Освітлення приміщення є одним з найважливіших факторів у роботі офісних працівників, оскільки вони майже весь свій час проводять за технікою. Освітлення приміщення можна поділити на природне та штучне. Природне освітлення – це освітлення, що потрапляє до приміщення від неба проходячи крізь вікна. Воно вважається найціннішим, до якого людське око пристосоване. З цього можна зробити висновок, що переважну більшість часу роботи працівник повинен проводити в приміщенні, де має використовуватись саме природне освітлення. Проте можна зіштовхнутись з рядом проблем. Цими проблемами можуть виступати:

- складність в рівномірному поширенні світла в приміщенні. Це означає, що неможливо забезпечити природнім світлом весь робочий простір однаково;
- хоча природне освітлення краще впливає на людину, ніж штучне, пряме влучення сонячних променів в органи зору діє негативно на організм людини;
- напрям природного освітлення залежить від часу доби. Це означає, що протягом 24 годин неможливо забезпечити однаковий рівень природного освітлення приміщення.

Штучне освітлення – освітлення, що реалізується шляхом використання штучних джерел світла. Використовуючи його можливо вирішити проблеми, що виникають при використанні лише природного освітлення.

Підвищена або понижена температура повітря робочої зони.

Оптимальною температурою вважається приблизно 20С°. Щоб досягти такої температури варто використовувати спеціальні пристрої для контролю температури та для її зміни до нормальної норми: кондиціонери, обігрівачі.

Підвищена або понижена вологість повітря.

Оптимальне значення вологості складає 25-60%. В даному випадку, коли приміщенням є офіс, більш вірогідно те, що вологість буде зменшена оскільки через роботу різної техніки буде виділятись тепло, що призведе до осушення повітря. Вирішити цю проблему можна шляхом встановлення в приміщені зволожувачів повітря та розпилювачів водяної пари.

Підвищене значення напруги в електричній мережі.

Електричний струм – це направлений рух заряджених частинок. Ще з початку винайдення, струм загрожує здоров’ю, життю та майну людини. Людина може отримати ураження електричним струмом через дотик до оголених проводів, через незнання правил безпеки при використовуванні техніки або через несправність приладів. Тож плануючи безпеку, що стосується електричного струму варто:

- ознайомити працівників з правилами безпеки та правилами поведінки під час роботи з технікою;
- обмежити контакт техніки та людини. Мається на увазі те, щоб людина контактувала з технікою безпечним шляхом, тобто взаємодіяла з нею використовуючи мишку та клавіатуру, і не контактуючи при цьому з вмістом системного блоку;
- в даному випадку кабель-менеджмент забезпечить безпеку самих працівників та зменшить кількість взаємодій між людиною і дротами;
- створення правильної проводки в приміщенні;
- найняття працівників, що будуть відповідальними за електрику в приміщенні. Це дозволить, не навантажувати працівників додатковою і зайвою роботою, так і зменшить ризик ураження електричним струмом.

Варто роздивитись кабель-менеджмент більш детально [22], оскільки він є достатньо важливим аспектом організації умов праці працівників. Організація дротів підвищить функціональність і комфорт при використанні техніки та її ремонті, за необхідності. Бюджетну організацію дротів можуть забезпечити хомути-стяжки, але вони мають декілька недоліків. Вони одноразові і їх використання не є досить практичним для офісу. На заміну їм прийшли липучки. Також гарною практикою стане використання різних патч-кордів, для створення «кольорового кодування». Використовуючи їх можна отримати гарно організований дата центр. Але всі ці рекомендації здебільшого стосуються серверної частини. Для прокладання дротів до персональних комп’ютерів чи просто офісних місць, варто використовувати спеціальні кабельні канали. Для реалізації гарної організації дротів та їх підключень до техніки та різного роду подовжувачів, варто використовувати кріплення та затискачі.

Недостатня освітленість робочої зони.

Під час роботи працівники майже весь день працюють з технікою і дивляться в монітор. Тому контроль яскравості екрану є досить важливим. Яскравість має залежати від освітлення приміщення. Воно має прямо пропорційно залежати від зовнішнього освітлення. Це означає, що під час світлої частини доби

варто підвищити яскравість, щоб добре бачити дрібні об’єкти. І навпаки, якщо вже темнішає, то і яскравість має зменшуватись. Також контроль освітлення може стосуватись не лише монітору. Для того щоб добре бачити прилади вводу, варто закуповувати техніку зі влаштованою підсвіткою. Це допоможе в темну частину доби добре бачити кнопки і не тільки. Не варто забувати про освітлення робочого місця за допомогою різного роду світильників. Концентрація уваги на одному об’єкті погано впливає на зір. Отже, варто робити певні вправи кожні пів години. Найпростішою вправою для зору є виконання однієї з наступних дій:

- тримаючи нерухомо голову та плечі, спрямовувати погляд вгору-вниз, вліво-вправо;
- переміщати очі за та проти годинниковою стрілкою або по діагоналях;
- декілька разів сфокусувати свій погляд на кінчику свого носа.

Організація власної робочої зони та дотримання правил користування технікою, що пов’язані зі здоров’ям працівника.

Під час роботи за комп’ютером у людини може почати боліти спина, або руки. Це відбувається тому, що вона знаходиться в неправильному положенні під час сидіння за комп’ютером. Є декілька способів вирішити таку проблему. Перший – надати людині зручне крісло для сидіння або/та комфортний стіл. В сучасному світі вони можуть бути різного типу, особливо корисними будуть ті які мають динамічну висоту, тобто можуть підніматись і опускатись, коли того захоче людина. Другий – людина може сама собі допомогти, а саме робити певні вправи кожен період часу, займати комфортну та корисну позу для сидіння.

Офісне приміщення не має великого спектру джерел виникнення пожежі, проте і він може чинити небезпеку для людського здоров’я і життя. Роздивимось кожен фактор більш детально.

Електрика.

Оскільки майже всі працівники мають електричну техніку для роботи, нею може бути комп’ютер, ноутбук або сервер, ризик виникнення пожежі досить

великий. Серед факторів виникнення пожежі, що пов’язані з електричним струмом, можна виділити найбільш впливові:

- погана проводка. Недбалість на етапі конструювання приміщення може призвести до короткого замикання в мережі, в результаті чого може виникнути пожежа;
- перевантаження електропроводки. Підключення до електричної мережі великої кількості техніки та сторонніх пристроїв, що власноруч створені, або неякісного виробництва, або ніяк не стосуються робочого процесу, може призвести до перегрівання провідника чи перенавантаження мережі;
- слабкий контакт в місцях з’єднання. В наслідок слабкого контакту може виникнути збільшення температури на місцях з’єднання. Це призведе до руйнування ізоляції і, як результат, може виникнути коротке замикання чи загорання тих об’єктів, що знаходяться поряд.

Оскільки причиною виникнення пожежі не завжди є людський фактор, варто роздивитись шляхи усунення не тільки з цього боку, а і зі сторони обладнання.

При виборі офісу варто враховувати якість проводки приміщення [23]. Для цього потрібно провести обстеження з залученням спеціаліста з електрики для виявлення пошкоджень та інших видів неполадок, з метою подальшого їх усунення перед початком робочого процесу. Також при налагодженні системи проводки варто враховувати будівельні матеріали, біля яких проходить дріт. Це означає, що гарною практикою буде проведення проводки в спеціальних захисних каналах.

Для усунення перегрівання та перевантаження мережі варто ознайомити працівників з правилами підключення пристроїв до електромережі офісу. Також приміщення має бути обладнане металевим чи з негорючого пластику електричним щитком. При його встановленні не варто забувати, що всі перемикачі та лічильники в ньому мають бути підключені професіоналом.

Людський фактор.

Встановивши найсучасніше обладнання для запобігання загоряння в офісі, все одно неможна виключити людський фактор, як причину пожежі. Серед великої кількості причин пов’язаних з людським фактором, можна виділити основні:

- недбале ставлення;
- незнання правил пожежної безпеки.

При влаштуванні працівника на робоче місце, потрібно провести інструктаж з техніки безпеки, ознайомити його з планом евакуації та правилами поводження з технікою в офісі. Для перевірки та закріплення знань, варто проводити два рази на рік навчальну тривогу.

Нажаль, встановлюючи найсучасніше обладнання та проводячи найрізноманітніші практики для робітників, все одно неможливо позбутися вірогідності виникнення пожежі. Тож важливим є необхідність визначити послідовність дій з детальним описом, які мають проводитися підчас виявлення пожежі.

Кожне приміщення повинно мати протипожежне обладнання для виявлення та ліквідації пожежі. Ними можуть бути вогнегасники, системи виявлення пожежі, вентиляції.

Не варто забувати про те, що для кожного приміщення має бути розроблена та затверджена керівництвом інструкція, яка має містити:

- план евакуації при надзвичайних ситуаціях;
- відведені місця для куріння;
- розпорядок проведення санітарних робіт та їх нормування.

Також підприємство повинно мати ряд документів, що описують дії при виявленні вогню. Ними можуть бути: правила відключення від мережі електричного обладнання; збірник правил, що нормує проведення інструктажів, та занять з працівниками для донесення інформації.

В даному розділі було розглянуто основні небезпечні фактори при роботі в офісному приміщенні. Небезпечними факторами можуть виступати: погана

освітленість; мікроклімат робочої зони; електричний струм; погана організація власної робочої зони. В результаті аналізу розглянутих факторів, були надані основні рекомендації щодо їх усунення. Небезпечні фактори варто враховувати, як на етапі вибору або конструюванні офісного приміщення, так і під час робочого процесу, щоб не допустити погіршення здоров’я працівників та виникнення надзвичайних ситуацій.

Особлива увага була приділена причинам виникнення та можливим варіантам усунення ризиків виникнення пожежі. Ризиками виникнення пожежі можуть виступати: погано організована електрика і недбале відношення до неї. В результаті детального дослідження ризиків, було надано основні рекомендації для усунення цих факторів, основні з яких це: правильне проектування приміщення, що буде схвалене спеціалістом та донесення до працівників правил пожежної безпеки та набору дій при виникненні пожежі.

ВИСНОВКИ

Результатом бакалаврської роботи є програмне застосування для прихованої передачі секретних даних за допомогою цифрових аудіо сигналів, для підвищення захисту інформації у відкритих каналах зв’язку, на основі модифікації стеганографічного методу, заснованого на сингулярному розкладі блоків матриці контейнера.

Для досягнення поставленої мети розв’язані наступні задачі.

Виконано огляд стеганографії, що працює з цифровими аудіо даними. Розглянуто області застосування стеганосистем. Сформована класифікація стеганографічних методів для занурення повідомлення в аудіо сигнали, на основі дослідженої літератури. Детально досліджено методи вбудовування повідомлення в області даних аудіо файлу: часову, частотну, частотно-часову, та методи вбудовування секретного повідомлення в службові області аудіо файлу: мета дані, проміжки між фреймами, кінець файлу.

Занурення інформації в службові частини файлу, мають значну перевагу в об’ємі інформації, що буде вбудована при відсутності явних змін даних аудіо файлу. Це означає, що при прослуховуванні такого файлу людина не помітить факт зміни файлу. Натомість програма, що перевірятиме значення полів у відповідності із значеннями формату, без проблем помітить втручання. Натомість методи, що пов’язані із областями даних, досить чутливі до певних видів атак чи змін, таких як стиснення.

Продемонстровані теоретичні основи модифікації стеганографічного алгоритму для цифрового аудіо сигналу.

Виконано детальне дослідження стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці цифрового зображення. Головною особливістю якого є стійкість до атак стиском, для деякого діапазону коефіцієнту стиску.

Представлені основні кроки модифікації стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці контейнера. Виконано

оцінку ефективності розробленого стеганографічного алгоритму в порівнянні з іншими стеганографічними програмними застосуваннями.

Представлена практична розробка модифікованого стеганографічного алгоритму, заснованого на сингулярному розкладі блоків матриці, для цифрового аудіо сигналу.

Описано середовище розробки, за допомогою якого було створено даний програмний продукт «Audio Steganography». Обґрунтовано, чому саме Matlab був вибраний, як середа для розробки програми.

Представлена організація функціонування програмного забезпечення. Для графічного відображення етапів роботи програми та демонстрації систематичної послідовності виконання поставлених задач, побудовані блок-схема етапу занурення секретного повідомлення та блок-схема вилучення секретного повідомлення.

Наведена інструкція користувача програмного забезпечення.

ПЕРЕЛІК ПОСИЛАНЬ

1. Карачка А.Ф. Технології захисту інформації. Тернопіль: ТНЕУ, 2017. 86 с.
2. Мінгальова Ю. Новітні криптографічні методи захисту інформації. *Науково-дослідна робота молодих учених: стан, проблеми, перспектив,* 2012. С.373-378.
3. Іванов В.Г. Захист інформації засобами комп'ютерної стеганографії. *Безпекове інноваційне суспільство: взаємодія у сфері правової освіти та правового виховання: матеріали міжнар. інтернет-конф,* 2016. С.53-56.
4. Юдін О.К., Зюбіна Р.В., Фролов О.В. Аналіз стеганографічних методів приховування інформаційних потоків у контейнери різних форматів. *Радиоэлектроника и информатика.* 2015. № 3. С.13-21.
5. Конахович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография. Теория и практика. К.: «МК-Пресс», 2006. 288 с.
6. Кодування звуку. *StudLancer*. URL: https://stud.com.ua/97175/ informatika/koduvannya_zvuku
7. Беспалов В.В. Информационные технологии. Томск: Изд-во Томского политехнического университета, 2011. 122 с.
8. Структура формата MP3. *Studbooks.* URL: https://studbooks.net/ 2183785 /informatika/struktura_formata
9. Внутри MP3. А как оно всё устроено? *Хабр.* URL: https://habr.com /ru/post/103635/
10. Сокрытие информации. Стеганография, Часть 2. *Информационная безопасность для всех.* URL: https://sec4all.ucoz.ru/publ/sokrytie_informacii_ steganografija_chast_2/1-1-0-66
11. Методы сокрытия информации в компьютерной стеганографии. *StudFiles.* URL: https://studfile.net/preview/10072102/page:28/
12. Методи стиснення з втратою даних. *Вікі ЦДПУ.* URL: https://wiki.cuspu.edu.ua/index.php/Методи_стиснення_з_втратою_даних_Ре шетник_2017

13. Забелин. М. А. Стегоанализ аудиоданных на основе методов сжатия. *Вестник СибГУТИ*. 2010. № 1. С.41-49.
14. Козіна М.О., Папковська О.Б., Логінова Н.І., Козін О.Б. Стеганоалгоритм, що використовує сингулярне розкладання матриці контейнера. *Сучасний захист інформації*. 2018. № 2. С.47-52.
15. Steganography. *GitHub.* URL: https://github.com/ragibson/Steganography#ByteSequenceManipulation
16. stegpy. *GitHub.* URL: https://github.com/dhsdshdhk/stegpy
17. Deepsound. *uptodonw*. URL: https://deepsound.ru.uptodown.com/windows
18. OpenPuff. *uptodonw.* URL: https://openpuff.ru.uptodown.com/windows
19. Конюшенко В.В. Начало работы с MATLAB. 2009. 73 с.
20. Дьяконов В. П. MATLAB 7.*/R2006/R2007: Самоучитель. М.: ДМК Пресс, 2008. 768 с.
21. Ануфриев И.Е., Смирнов А. Б., Смирнова Е. Н. MATLAB 7. СПб.: БХВ-Петербург, 2005. 1104 с.
22. Рай перфекциониста или каким должен быть кабель-менеджмент. *Хабрахабр*. URL: https://habr.com/ru/company/ruvds/blog/320610/
23. Жидецький В.Ц. Охорона праці користувачів комп’ютерів. Львів, Афіша, 2000. 176 с.

Додаток А

Лістинг програмного коду

```
function varargout = VoiceDetection(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @VoiceDetection_OpeningFcn, ...
                   'gui_OutputFcn',  @VoiceDetection_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% --- Executes just before VoiceDetection is made visible.
function VoiceDetection_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
clean_load(hObject, handles);
guidata(hObject, handles);

function clean_load(hObject, handles)
global signal_struct;
signal_struct{1} = []; % signal
signal_struct{2} = []; % time
signal_struct{3} = 0;  % fs
signal_struct{4} = 0;  % bits
signal_struct{5} = [];  % secret
signal_struct{6} = [];  % stego
signal_struct{7} = [];  % stegomat

axes(handles.axes_signal);
cla(handles.axes_signal);
set(handles.axes_signal,'XGrid','on');
set(handles.axes_signal,'YGrid','on');
set(get(handles.axes_signal,'XLabel'),'String','Час','FontSize',10);
set(get(handles.axes_signal,'YLabel'),'String','Амплітуда','FontSize
',10);
set(handles.axes_signal,'YLim',[-1.1 1.1]);

axes(handles.axes_signal_stego);
cla(handles.axes_signal_stego);
set(handles.axes_signal_stego,'XGrid','on');
set(handles.axes_signal_stego,'YGrid','on');
```

```
set(get(handles.axes_signal_stego,'XLabel'),'String','Час','FontSize
',10);
set(get(handles.axes_signal_stego,'YLabel'),'String','Амплітуда','Fo
ntSize',10);
set(handles.axes_signal_stego,'YLim',[-1.1 1.1]);

set(handles.edit_name,'String','');
set(handles.edit_bits,'String','');
set(handles.edit_frequency,'String','');
set(handles.edit_channels,'String','');
set(handles.edit_duration,'String','');
set(handles.edit_size,'String','');

guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = VoiceDetection_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;

% --- Executes on button press in btn_load_file.
function btn_load_file_Callback(hObject, eventdata, handles)
global signal_struct;
clean_load(hObject, handles);

[filename,pathname]=uigetfile({'*.flac';'*.wav';'*.ogg'},'Select
file');
if(isequal(filename,0))
else
    [x,fs]=audioread([pathname filename]);
    signal_struct{1} = x;
    signal_struct{3} = fs;

    info=audioinfo([pathname filename]);
    set(handles.edit_name,'String',filename);

    if(isfield(info,'BitsPerSample'))
        set(handles.edit_bits,'String',[num2str(info.BitsPerSample)
'b']);
        signal_struct{4} = info.BitsPerSample;
    end

    set(handles.edit_frequency,'String',[num2str(signal_struct{3})
'Hz']);

    if(isfield(info,'NumChannels'))

set(handles.edit_channels,'String',num2str(info.NumChannels));
    end

    if(isfield(info,'Duration'))
        set(handles.edit_duration,'String',[num2str(info.Duration)
's']);
```

```
    end

    size=dir([pathname filename]);
    set(handles.edit_size,'String',[num2str(size.bytes) 'b']);

    axes(handles.axes_signal);
    cla(handles.axes_signal);
    signal_struct{2}(:,1) = [0:(length(signal_struct{1})-
1)]/signal_struct{3};
    plot(handles.axes_signal,signal_struct{2},signal_struct{1},'b');
    set(handles.axes_signal,'XGrid','on');
    set(handles.axes_signal,'YGrid','on');

set(get(handles.axes_signal,'XLabel'),'String','Час','FontSize',10);

set(get(handles.axes_signal,'YLabel'),'String','Амплітуда','FontSize
',10);
    set(handles.axes_signal,'YLim',[-1.1 1.1]);

    guidata(hObject, handles);
end

% --- Executes on button press in btn_listen.
function btn_listen_Callback(hObject, eventdata, handles)
global signal_struct;
if(isempty(signal_struct{1}))
    msgbox('Введіть сигнал');
else
    sound(signal_struct{1},signal_struct{3});
end

% --- Executes on button press in btn_load_micro.
function btn_load_micro_Callback(hObject, eventdata, handles)
clean_load(hObject, handles);
set(handles.edit_name,'String','test.wav');
set(handles.edit_bits,'String','24b');
set(handles.edit_frequency,'String','16000Hz');
set(handles.edit_channels,'String','1');
set(handles.edit_duration,'String','4s');
set(handles.edit_size,'Enable','off');

VoiceRecorder(hObject,handles);

function []=VoiceRecorder(hObject,handles)
hF=figure('Name','Рекордер','NumberTitle','off','MenuBar','none');
set(hF,'Position',[500 500 200 70]);
hbtn_start=uicontrol('Style','pushbutton','position',[10 20 60
30],'String','Старт','CallBack',{@hbtn_startCallback,hObject,handles
});
hbtn_stop=uicontrol('Style','pushbutton','position',[70 20 60
30],'String','Стоп','CallBack',{@hbtn_stopCallback,handles});
```

```
hbtn_save=uicontrol('Style','pushbutton','position',[130 20 60
30],'String','Зберегти','CallBack',{@hbtn_saveCallback,hObject,handl
es});

function hbtn_startCallback(src,evt,hObject, handles)
global flag;
flag = 1;
fs = get(handles.edit_frequency,'String');
fs = str2double(fs(1:end-2));
bits = get(handles.edit_bits,'String');
bits = str2double(bits(1:end-1));
channels = str2double(get(handles.edit_channels,'String'));
record = audiorecorder(fs,bits,channels);
duration = get(handles.edit_duration,'String');
duration = str2double(duration(1:end-1));
buffer_size = ceil(duration*fs);
buffer(1:buffer_size,1) = 0;
time_frame = 0.8;
time_total = buffer_size/fs;
time_step = time_total/buffer_size;
time(:,1) = -time_total+time_step:time_step:0;

axes(handles.axes_signal);
while(1)
    if(flag)
        recordblocking(record,time_frame);
        temp_buffer = getaudiodata(record);
        [n,m] = size(temp_buffer);

        buffer(1:end-n) = buffer(n+1:end);
        buffer(end-n+1:end) = temp_buffer;

        time = [time(n+1:end,1);time(end,1)+[1:n]'*time_step];

        plot(handles.axes_signal,time,buffer,'b');
        set(handles.axes_signal,'XGrid','on');
        set(handles.axes_signal,'YGrid','on');

set(get(handles.axes_signal,'XLabel'),'String','Час','FontSize',10);

set(get(handles.axes_signal,'YLabel'),'String','Амплітуда','FontSize
',10);
        set(handles.axes_signal,'YLim',[-1.1 1.1]);
        set(handles.axes_signal,'XLim',[min(time) max(time)]);
    else
        pause(0.1);
        break;
    end
end

global signal_struct;
signal_struct{1} = buffer;
signal_struct{3} = fs;
```

```
time_total = length(signal_struct{1})/signal_struct{3};
time_step = time_total/length(signal_struct{1});
test(:,1)= 0:time_step:time_total-time_step;
signal_struct{2} = test;
guidata(hObject,handles);

function hbtn_stopCallback(src,evt,handles)
global flag;
flag = 0;

function hbtn_saveCallback(src,evt,hObject,handles)
global signal_struct;
if(isempty(signal_struct{1}))
    msgbox('Запишить сигнал');
else
    rez = questdlg('Зберегти сигнал в файлі?','','Так','Ні','Так');
    if(strcmp(rez,'Так'))
        filename = get(handles.edit_name,'String');
        bits = get(handles.edit_bits,'String');
        bits = bits(1:end-1);
        bits = str2double(bits);

audiowrite(filename,signal_struct{1},signal_struct{3},'BitsPerSample
',bits);
        set(handles.edit_size,'Enable','on');
        size = dir(filename);
        set(handles.edit_size,'String',[num2str(size.bytes) 'b']);
        msgbox(['Сигнал збережено в ' filename]);
    end
end

guidata(hObject,handles);
close Рекордер;

% --- Executes on button press in btn_gener_secret_bin.
function btn_gener_secret_bin_Callback(hObject, eventdata, handles)
global signal_struct;
vec_sound=signal_struct{1};
secret=[];
if(~isempty(vec_sound))
    N=8;
    n=fix(size(vec_sound,1)/(N*N));
    m=n-32;
    secret(:,1) = de2bi(m,32);
    signal_struct{5}=[secret;round(rand(m,1))];
    guidata(hObject,handles);
    msgbox('Повідомлення згенеровано');
else
    msgbox('Завантажте контейнер');
end

% --- Executes on button press in btn_gener_secret_text.
function btn_gener_secret_text_Callback(hObject, eventdata, handles)
```

```
prompt = {'Секретне повідомлення:'};
dlgtitle = 'Введіть';
dims = [5 35];
definput = {'secret',};
answer = inputdlg(prompt,dlgtitle,dims,definput);

secret=[];
if(~isempty(answer{1,1}))
    input=answer{1,1};
    input_size=size(input,2);

    global signal_struct;
    vec_sound=signal_struct{1};
    if(~isempty(vec_sound))
        N=8;
        n=fix(size(vec_sound,1)/(N*N));
        if(input_size*N<4294967295 && input_size*N<n-32)
            secret(:,1) = de2bi(input_size,32);
            for i=1:input_size
                temp1=input(i);
                temp2=double(temp1);
                temp3(:,1)=de2bi(temp2,8);
                secret=[secret;temp3];
            end
            signal_struct{5}=secret;
            guidata(hObject,handles);
            msgbox('Повідомлення згенеровано');
        else
            msgbox('Повідомлення занадто велике');
        end
    else
        msgbox('Завантажте контейнер');
    end
end

% --- Executes on button press in btn_input_secret.
function btn_input_secret_Callback(hObject, eventdata, handles)
global signal_struct;
x = signal_struct{1};
fs = signal_struct{3};
if(isempty(x))
    msgbox('Завантажте контейнер');
else
    secret = signal_struct{5};
    if(isempty(secret))
        msgbox('Завантажте повідомлення');
    else
        vec_sound=x(:,1);
        N=8;
        n=fix(size(vec_sound,1)/(N*N));
        vec_sound=vec_sound(1:n*N*N);
        vec_sound=round(((vec_sound+1)/2)*255);
        cbx_g=get(handles.cbx_gor,'Value');
```

```
        cbx_v=get(handles.cbx_ver,'Value');
        cbx_d=get(handles.cbx_diag,'Value');
        if(cbx_g==1)
            mat_sound=ConvertG(vec_sound,N);
        else
            if(cbx_v==1)
                mat_sound=ConvertV(vec_sound,N);
            else
                mat_sound=ConvertD(vec_sound,N);
            end
        end
        [mat_sound_steg,L] = Input(mat_sound, secret, N);
        if(cbx_g==1)
            vec_sound_steg=UConvertG(mat_sound_steg,N);
        else
            if(cbx_v==1)
                vec_sound_steg=UConvertV(mat_sound_steg,N);
            else
                vec_sound_steg=UConvertD(mat_sound_steg,N);
            end
        end
        vec_sound_steg=(((vec_sound_steg/255)*2)-1);
        x(1:size(vec_sound_steg,1),1)=vec_sound_steg;
        signal_struct{6}=x;
        signal_struct{7}=mat_sound_steg;
        guidata(hObject,handles);

        axes(handles.axes_signal_stego);
        cla(handles.axes_signal_stego);
        signal_struct{2}(:,1) = [0:(length(signal_struct{1})-
1)]/signal_struct{3};

plot(handles.axes_signal_stego,signal_struct{2},signal_struct{6},'b'
);
        set(handles.axes_signal_stego,'XGrid','on');
        set(handles.axes_signal_stego,'YGrid','on');

set(get(handles.axes_signal_stego,'XLabel'),'String','Час','FontSize
',10);

set(get(handles.axes_signal_stego,'YLabel'),'String','Амплітуда','Fo
ntSize',10);
        set(handles.axes_signal_stego,'YLim',[-1.1 1.1]);
        msgbox('Занурення виконано');
    end
end

% --- Executes on button press in btn_listen_stego.
function btn_listen_stego_Callback(hObject, eventdata, handles)
global signal_struct;
if(isempty(signal_struct{6}))
    msgbox('Виконайте занурення повідомлення');
else
```

```
    sound(signal_struct{6},signal_struct{3});
end

% --- Executes on button press in btn_save_stego.
function btn_save_stego_Callback(hObject, eventdata, handles)
global signal_struct;
if(isempty(signal_struct{6}))
    msgbox('Виконайте занурення повідомлення');
else
    [file,path] = uiputfile('test.flac');
    if isequal(file,0) || isequal(path,0)
        msgbox('Файл не збережено');
    else
        path=fullfile(path,file);
        global signal_struct;
        audiowrite(path,signal_struct{6},signal_struct{3});
        msgbox(['Сигнал збережено в ' file]);
    end
end

% --- Executes on button press in btn_analyze_stego.
function btn_analyze_stego_Callback(hObject, eventdata, handles)
global signal_struct;
if(isempty(signal_struct{6}))
    msgbox('Виконайте занурення повідомлення');
else
    N=8;
    block_size=size(signal_struct{7},1)/N;
    prompt = {['Номер блоку від 1 до ' num2str(block_size)]};
    dlgtitle = 'Введіть';
    dims = [1 35];
    definput = {'1',};
    answer = inputdlg(prompt,dlgtitle,dims,definput);
    if(isempty(answer))
        msgbox('Введіть номер блоку');
    else
        num=str2num(answer{1,1});
        matrix=signal_struct{7};
        test=matrix((num-1)*N+1:num*N,:);
        [U,D,V]=svd(test);
        figure
        subplot(1,2,1), imshow(uint8(test));
        subplot(1,2,2),
        plot(diag(D));
        axis square;
        grid on
        xlabel('№');
        ylabel('SV');
    end
end

% --- Executes on button press in btn_get_secret.
function btn_get_secret_Callback(hObject, eventdata, handles)
```

```
global signal_struct;
[filename,pathname]=uigetfile({'*.flac';'*.wav';'*.ogg'},'Select
file');
if(~isequal(filename,0))
    [x,fs]=audioread([pathname filename]);
    vec_sound=x(:,1);
    N=8;
    n=fix(size(vec_sound,1)/(N*N));
    vec_sound=vec_sound(1:n*N*N);
    vec_sound=round(((vec_sound+1)/2)*255);
    cbx_g=get(handles.cbx_gor,'Value');
    cbx_v=get(handles.cbx_ver,'Value');
    cbx_d=get(handles.cbx_diag,'Value');
    if(cbx_g==1)
        mat_sound_steg=ConvertG(vec_sound,N);
    else
        if(cbx_v==1)
            mat_sound_steg=ConvertV(vec_sound,N);
        else
            mat_sound_steg=ConvertD(vec_sound,N);
        end
    end
    vector_bin = Output(signal_struct{7},N);
    secret = ConvertStego(vector_bin);
    msgbox(secret);
end

function [M]=ConvertV(V,N)
M=[];
for i=1:N*N:size(V,1)
    T=V(i:i+N*N-1,1);
    TT=[];
    for j=1:N:size(T,1)
        TT=[TT T(j:j+N-1)];
    end
    M=[M;TT];
end

function [M]=ConvertG(V,N)
M=[];
for i=1:N*N:size(V,1)
    T=V(i:i+N*N-1,1);
    TT=[];
    for j=1:N:size(T,1)
        TT=[TT; (T(j:j+N-1))'];
    end
    M=[M;TT];
end

function [M]=ConvertD(V,N)
M=[];
for i=1:N*N:size(V,1)
    T=V(i:i+N*N-1,1);
```

```
    TT=[];
    X=[1  3  4   10  11 21 22 36;
       2  5  9   12  20 23 35 37;
       6  8  13 19  24 34 38 49;
       7 14 18  25 33 39 48 50;
       15 17 26 32 40 47 51 58;
       16 27 31 41 46 52 57 59;
       28 30 42 45 53 56 60 63;
       29 43 44 54 55 61 62 64];

    for j=1:N*N
        [k,t]=find(X==j);
        TT(k,t)=T(j);
    end
    M=[M;TT];
end

function [V]=UConvertV(M,N)
V=[];
for i=1:N:size(M,1)
    T=M(i:i+N-1,1:N);
    TT=[];
    for j=1:size(T,2)
        TT=[TT;T(:,j)];
    end
    V=[V;TT];
end

function [V]=UConvertG(M,N)
V=[];
for i=1:N:size(M,1)
    T=M(i:i+N-1,1:N);
    TT=[];
    for j=1:size(T,2)
        TT=[TT;(T(j,:))'];
    end
    V=[V;TT];
end

function [V]=UConvertD(M,N)
V=[];
for i=1:N:size(M,1)
    T=M(i:i+N-1,1:N);
    TT=[];
    X=[1  3  4   10  11 21 22 36;
       2  5  9   12  20 23 35 37;
       6  8  13 19  24 34 38 49;
       7 14 18  25 33 39 48 50;
       15 17 26 32 40 47 51 58;
       16 27 31 41 46 52 57 59;
       28 30 42 45 53 56 60 63;
       29 43 44 54 55 61 62 64];
```

```
    for j=1:N*N
        [k,t]=find(X==j);
        TT=[TT;T(k,t)];
    end
    V=[V;TT];
end

function [B,L] = Input(A, M, N)

L=[];
k=1;
for i=1:N:size(M,1)*N
    for j=1:N:size(A,2)
        T=A(i:i+N-1,j:j+N-1);
        [U,S,V]=svd(T);

        L(k,1)=S(1,1);
        if (S(1,1)>=100)
            temp=(fix(S(1,1)/100))*100;
            L(k,2)=(fix(S(1,1)/100))*100;
        else
            temp=(fix(S(1,1)/10))*10;
            L(k,2)=(fix(S(1,1)/10))*10;
        end

        ns=0;
        if(M(k,1)==0)
            ns=temp+0.25*S(2,2);
        end

        if(M(k,1)==1)
            ns=temp+0.75*S(2,2);
        end

        L(k,3)=temp+0.25*S(2,2);
        L(k,4)=temp+0.75*S(2,2);
        L(k,5)=S(2,2);

        S(1,1)=ns;
        TT= (U*S*(V)');
        B(i:i+N-1,j:j+N-1)=TT;
        k=k+1;
    end
end

function [R] = Output(B,N)
R=zeros(size(B,1)/N,size(B,2)/N);
for i=1:N:size(B,1)
    for j=1:N:size(B,2)
        T=B(i:i+N-1,j:j+N-1);
        [~,S,~]=svd(T);

        temp=0;
```

```
        if (S(1,1)>=100)
            temp=(fix(S(1,1)/100))*100;
        else
            temp=(fix(S(1,1)/10))*10;
        end

         test=S(1,1)-temp;
         if(test<0.5*(S(2,2)))
             R((i-1)/N+1,(j-1)/N+1)=0;
         else
             R((i-1)/N+1,(j-1)/N+1)=1;
         end
    end
end

function [rez] = ConvertStego(V)
N=8;
size_secret = V(1:32,1);
size_secret = bi2de(size_secret');
secret=V(33:end,1);
rez=[];
for i=1:size_secret
    test1=secret((i-1)*N+1:i*N,1);
    test2=bi2de(test1');
    test3=char(test2);
    rez=[rez test3];
end
```