Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Підвищення стійкості методу приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування у частотній області та шифрування ключ-блоку»

Виконав: здобувач 2-го курсу,
групи 2КІТС-24м
спеціальності 125– Кібербезпека
та захист інформації
Освітня програма – Кібербезпека
інформаційних технологій та систем

Олексюк Є. М.

Керівник:
д.т.н., проф., проф. каф. МБІС
Яремчук Ю. Є.
«09» грудня 2025 р.

Опонент: д.ф., доц., доц. каф. ОТ
Обертюх М. Р.
«09» грудня 2025 р.

Допущено до захисту
Голова секції УБ кафедри МБІС

Юрій ЯРЕМЧУК
“09” грудня 2025 р.

Вінниця ВНТУ - 2025 рік

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

Рівень вищої освіти ІІ-й (магістерський)
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека та захист інформації
Освітньо-професійна програма - Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ
Голова секції УБ, кафедра МБІС

_____ **Юрій ЯРЕМЧУК**
“__24__” ___вересня___ 2025 р.

З А В Д А Н Н Я
на магістерську кваліфікаційну роботу студенту
Олексюку Євгенію Михайловичу

1. Тема роботи «Підвищення стійкості методу приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування у частотній області та шифрування ключ-блоку» .
Керівник роботи д.т.н., проф., проф. каф. МБІС Яремчук Юрій Євгенович .

затверджені наказом вищого навчального закладу від “24” вересня 2025 року № 313
2. Строк подання студентом роботи 01.12.2025 р.
3. Вихідні дані до роботи: Методичні вказівки до виконання магістерської кваліфікаційної роботи; наукові публікації та монографії з питань цифрової стеганографії та обробки аудіосигналів; математичний опис методу QIM; алгоритми симетричного шифрування для захисту ключ-блоку.
4. Зміст текстової частини:
1. Аналітичний огляд сучасних методів приховування даних в аудіосигналах
2. Розроблення методу підвищення стійкості приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування та шифрування ключ-блоку
3. Програмна реалізація методу підвищення стійкості приховування даних в аудіосигналах
4. Економічна частина
5. Перелік ілюстративного матеріалу (з точним зазначенням обов`язкових креслень)
Схеми класифікації стеганографічних методів; структурна схема запропонованого методу вбудовування та вилучення даних; блок-схеми алгоритмів шифрування ключ-блоку; графіки залежності ймовірності бітової помилки (BER) від бітрейту

MP3-стиснення; спектрограми аудіосигналів до та після вбудовування; діа
економічної ефективності проекту; презентація у форматі PowerPoint.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина	Яремчук Ю. Є., д.т.н., проф. каф. МБІС		
Економічна частина	Ратушняк О. Г., к.т.н., доц. каф. ЕПВМ		

7. Дата видачі завдання 24 вересня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи		Примітка
1	Отримання завдання, підбір та аналіз літературних джерел за темою дослідження	24.09.2025	05.10.2025	
2	Аналіз існуючих рішень та обгрунтування вибору методу дослідження	06.10.2025	15.10.2025	
3	Розробка математичної моделі та алгоритму вбудовування даних з шифруванням ключ-блоку	16.10.2025	28.10.2025	
4	Програмна реалізація методу та проведення експериментів	29.10.2025	14.11.2025	
5	Підготовка економічної частини	15.11.2025	19.11.2025	
6	Оформлення пояснювальної записки, підготовка графічного матеріалу та презентації	20.11.2025	26.11.2025	
7	Переддипломний захист			
8	Захист магістерської кваліфікаційної роботи			

Студент _____ Олексюк Є. М.
(підпис)

Керівник роботи _____ Яремчук Ю. Є.
(підпис)

АНОТАЦІЯ

УДК 004.056.5

Олексюк Є.М. Підвищення стійкості методу приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування у частотній області та шифрування ключ-блоку. Магістерська кваліфікаційна робота зі спеціальності 125 – Кібербезпека, освітня програма – Кібербезпека інформаційних технологій та систем. Вінниця: ВНТУ, 2025. – 86 с.

На укр. мові. Бібліогр.: 34 назв; рис.: 23; табл.: 22.

Магістерська кваліфікаційна робота присвячена розробці методу підвищення стійкості приховування даних в аудіосигналах до стиснення MP3. В рамках роботи проведено аналітичний огляд сучасних методів аудіостеганографії, визначено їх переваги та недоліки. Виконано порівняльний аналіз стійкості існуючих методів до MP3-компресії. Розроблено гібридний метод на основі QIM-квантування у частотній області MDCT з додатковим криптографічним захистом ключ-блоку за допомогою AES-256. Створено програмну реалізацію методу на мові Python з використанням бібліотек NumPy, SciPy, librosa та cryptography. Виконано тестування програмного засобу та аналіз результатів.

Ілюстративна частина складається з 12 плакатів.

В економічному розділі оцінено витрати на розробку технології та програмного засобу.

Ключові слова: аудіостеганографія, QIM-квантування, MP3-компресія, шифрування ключ-блоку, MDCT, кібербезпека, приховування даних.

ABSRTACT

UDC 004.056.5

Oleksiuk Y. M. Improving the robustness of a data hiding method in audio signals against MP3 compression based on QIM quantization in the frequency domain and key-block encryption. Master's thesis in specialty 125 – Cybersecurity, educational program – Cybersecurity of Information Technologies and Systems. Vinnytsia: VNTU, 2025. – 86 p.

In Ukrainian. Bibliographer: 34 titles; fig.: 23; tabl.: 22.

The master's thesis is devoted to developing a method for enhancing the robustness of data hiding in audio signals under MP3 compression. The work includes an analytical review of modern audio steganography methods, identifying their advantages and limitations. A comparative analysis of the robustness of existing methods against MP3 compression has been carried out. A hybrid method has been developed based on QIM quantization in the MDCT frequency domain with additional cryptographic protection of the key block using AES-256. A software implementation of the method was created in Python using the NumPy, SciPy, librosa, and cryptography libraries. Software testing and result analysis have been performed.

The illustrative part consists of 12 posters.

The economic section evaluates the costs of developing the technology and software tool.

Keywords: audio steganography, QIM quantization, MP3 compression, key-block encryption, MDCT, cybersecurity, data hiding.

ЗМІСТ

ВСТУП

В умовах стрімкого розвитку інформаційно-комунікаційних технологій проблема забезпечення конфіденційності та цілісності даних набуває особливої ваги. Традиційні методи криптографічного захисту, хоч і забезпечують високий рівень безпеки, часто привертають увагу зловмисників самим фактом наявності зашифрованого контенту. У цьому контексті стеганографія — наука про приховування факту передавання інформації — виступає ефективним додатковим засобом захисту даних.

Серед різних типів носіїв інформації особливе місце займають аудіосигнали, які широко використовуються у мультимедійних комунікаціях, потокових сервісах і системах цифрового захисту авторських прав. Проте значна частина аудіоконтенту піддається стисненню у форматі MP3, що істотно впливає на вбудовану приховану інформацію, руйнуючи або спотворюючи її. Тому проблема підвищення стійкості стеганографічних методів до стиснення MP3 залишається актуальною і потребує нових підходів.

Одним із перспективних напрямів є використання методів квантування з вбудовуванням інформації (Quantization Index Modulation, QIM) у частотній області аудіосигналу. Ці методи забезпечують високий баланс між непомітністю, стійкістю та пропускною здатністю. Для додаткового підвищення безпеки може застосовуватися шифрування ключ-блоку, що керує псевдовипадковим порядком вставки бітів повідомлення. Такий підхід дозволяє ускладнити несанкціонований доступ і збільшує криптостійкість системи.

Таким чином, розроблення гібридного методу приховування даних в аудіосигналах, стійкого до стиснення MP3, є важливим науковим та практичним завданням у галузі кібербезпеки.

Об'єктом дослідження бакалаврської роботи є процес приховування даних в аудіосигналах із використанням методів частотного перетворення та квантування.

Предметом дослідження є методи та алгоритми підвищення стійкості прихованої інформації до втрат, спричинених стисненням MP3, з використанням QIM-квантування та шифрування ключ-блоку.

Мета цього дослідження полягає у підвищенні стійкості методу приховування даних в аудіосигналах до стиснення MP3 шляхом використання QIM-квантування у частотній області та шифрування ключ-блоку.

У роботі використовуються методи цифрової обробки сигналів (ДПФ, ДХП), теорія квантування, елементи криптографії (блокове шифрування), теорія інформації, методи моделювання, статистичного аналізу та програмного тестування. Реалізація експериментів здійснюється з використанням середовищ Python та MATLAB/Octave.

Наукова новизна отриманих результатів, запропоновано комбінований метод приховування даних в аудіосигналах, який поєднує QIM-квантування у частотній області та динамічне шифрування ключ-блоку для керування псевдовипадковим розподілом вбудованих бітів.

Запропонований підхід дозволяє підвищити стійкість вбудованої інформації до стиснення MP3 без істотного зниження якості аудіосигналу.

Практичне значення отриманих результатів, розроблений метод може бути використаний у системах цифрового водяного маркування для захисту авторських прав, прихованих каналах зв’язку та безпечній передачі службових даних, системах DRM (Digital Rights Management) та корпоративних аудіоплатформах.

Розроблене програмне забезпечення може слугувати основою для побудови прикладних стеганографічних систем у галузі кібербезпеки.

1 АНАЛІТИЧНИЙ ОГЛЯД СУЧАСНИХ МЕТОДІВ ПРИХОВУВАННЯ ДАНИХ В АУДІОСИГНАЛАХ

1.1 Сучасні методи приховування інформації в аудіосигналах

Приховування інформації в аудіосигналах (аудіостеганографія) є важливим напрямом захисту інформації, який дозволяє здійснювати передачу даних у прихованому вигляді через акустичні канали. Основна мета таких методів полягає у внесенні незначних змін до параметрів аудіосигналу таким чином, щоб приховане повідомлення було неможливо виявити на слух або стандартними методами аналізу сигналу.

Залежно від принципу модифікації носія, методи приховування інформації в аудіосигналах поділяються на три основні групи:

1. методи у часовій області;
2. методи у частотній області;
3. комбіновані (гібридні) методи, що поєднують обидва підходи.

Методи часової області є найпростішими за реалізацією. Найбільш відомий серед них — LSB (Least Significant Bit), який полягає у зміні наймолодших бітів амплітуди аудіосемплів відповідно до бітів прихованого повідомлення (формула 1.1). Формально, якщо x_i— амплітуда і-го семплу, то модифікація має вигляд:

$$x_i' = (x_i \text{ AND } 11111110)_2 \text{ OR } m_i \qquad (1.1)$$

де m_i— біт повідомлення.

Перевагою LSB-методу є висока пропускна здатність та простота реалізації. Недоліком — низька стійкість до перетворень сигналу (стиснення, фільтрації, реверсу), що робить цей метод малопридатним для практичного використання у сучасних умовах [1].

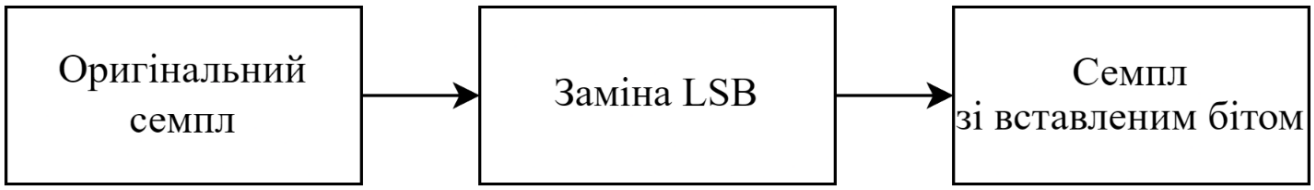


Рисунок 1.1 – Схематичне зображення процесу вставки біта повідомлення у методі LSB

У сучасних дослідженнях основну увагу приділено методам частотної області, які забезпечують вищу стійкість до обробки аудіо (зокрема до MP3-компресії). Вони використовують трансформації сигналу — дискретне косинусне (DCT), дискретне вейвлетне (DWT) або дискретне Фур'є перетворення (DFT).

Одним з найбільш ефективних підходів є метод квантування з інформаційною модуляцією (QIM – Quantization Index Modulation), запропонований Chen і Wornell [2]. Цей метод став фундаментальним у цифровому водяному знакуванні та стеганографії, оскільки дозволяє досягати високої стійкості до атак та компресії при мінімальних спотвореннях носія. Суть методу полягає у модифікації індексів квантування частотних коефіцієнтів відповідно до бітів повідомлення, яке необхідно вбудувати. Для цього створюються дві або більше решіток квантування, кожна з яких відповідає певному біту даних (наприклад, 0 або 1).

У загальному вигляді квантування для вбудовування повідомлення m визначається (формула 1.2):

$$y = Q_m(x) = \Delta \cdot \text{round}\left(\frac{x - d_m}{\Delta}\right) + d_m \quad (1.2)$$

де

- y— модифікований коефіцієнт сигналу;
- x— вихідний коефіцієнт у частотній області;
- Δ— крок квантування;
- d_m— зсув квантувальної решітки, який залежить від біта повідомлення m.

Таким чином, для кожного біта повідомлення використовується окрема решітка квантування, що дозволяє вбудовувати інформацію без суттєвого збільшення похибки відновлення сигналу.

Під час вилучення інформації приймач виконує зворотну операцію – визначає, до якої решітки належить прийняте значення коефіцієнта y. Ця процедура не вимагає знання самого сигналу-носія, що є значною перевагою QIM перед класичними методами, зокрема LSB-заміною.

Переваги QIM можна назвати:

– висока стійкість до втрат і стиснення, завдяки роботі у частотному просторі, QIM добре витримує втрати при MP3-компресії, оскільки зміни квантувальних рівнів зберігаються навіть після кодування втрат.контрольована спотворюваність сигналу;

– прозорість вбудовування, малі зміни у частотних коефіцієнтах не викликають помітних спотворень при відтворенні аудіосигналу.

– математична строгість, модель QIM базується на чітко визначених операціях квантування, що полегшує формальний аналіз спотворень та стійкості.

Для забезпечення додаткової безпеки у сучасних реалізаціях QIM застосовується ключова модуляція, коли вибір конкретної решітки визначається не лише бітом повідомлення, але й секретним ключем. Це значно ускладнює спроби несанкціонованого вилучення даних або виявлення самого факту стеганографічного вбудовування.

На практиці метод QIM часто реалізують у частотній області, після перетворення аудіосигналу за допомогою MDCT (Modified Discrete Cosine Transform) або DWT (Discrete Wavelet Transform). Це дозволяє вибирати спектральні області, менш чутливі до компресії, і підвищувати стійкість прихованих даних до MP3-кодування.

Комбіновані (гібридні) методи поєднують переваги часової та частотної областей. Наприклад, у Abdallah H. A., Meshoul S. A [3] запропоновано використання DWT + QIM, де вейвлет-перетворення забезпечує адаптивний

розподіл енергії сигналу, а QIM відповідає за вбудовування бітів з контрольованою стійкістю.

Інший метод часової області — Echo Hiding, коли дані вбудовуються у вигляді слабкого ехо-сигналу із контрольованими параметрами затримки. Такий метод має високу прихованість, але обмежену пропускну здатність [4].

Таблиця 1.1 – Порівняльний аналіз сучасних методів аудіостеганографії

Метод	Область	Пропускна здатність	Стійкість до MP3	Помітність	Складність реалізації
LSB	часова	висока	низька	низька	низька
Echo Hiding	часова	середня	середня	дуже низька	середня
DCT-квантування	частотна	середня	висока	низька	середня
QIM	частотна	середня	висока	низька	висока
DWT + QIM	комбінована	середня	дуже висока	низька	висока

Проведений аналіз показав, що методи часової області, такі як LSB, поступаються за стійкістю сучасним частотним підходам. Методи на основі QIM-квантування є найбільш перспективними для приховування даних у реальних умовах, зокрема при стисненні у форматі MP3. Поєднання QIM з блоковим шифруванням ключа дозволяє підвищити як безпеку, так і контроль над процесом вбудовування.

1.2 Проблеми стійкості методів приховування даних до стиснення MP3

Однією з ключових проблем сучасних методів приховування інформації в аудіосигналах є нестійкість до стиснення, зокрема при перетворенні у формат MP3 (MPEG-1 Layer III). Формат MP3 широко використовується для зменшення розміру звукових файлів без значної втрати якості звучання, однак його особливість

полягає у психоакустичному видаленні малозначущих компонентів сигналу, які часто використовуються для вбудовування прихованої інформації [5].

Алгоритм MP3 складається з кількох етапів:

1. Перетворення сигналу з часової області у частотну за допомогою модифікованого дискретного косинусного перетворення (MDCT);

2. Психоакустичне моделювання, яке визначає частотні ділянки, непомітні для людського слуху (маскування звуків);

3. Квантування і кодування Хаффмана, що дозволяє суттєво зменшити обсяг даних.

Під час цього процесу значна частина високочастотних і малопомітних компонентів сигналу відкидається, що призводить до знищення або спотворення прихованих бітів, якщо вони були вставлені у такі частини сигналу [6].

Ілюстрація впливу стиснення MP3:

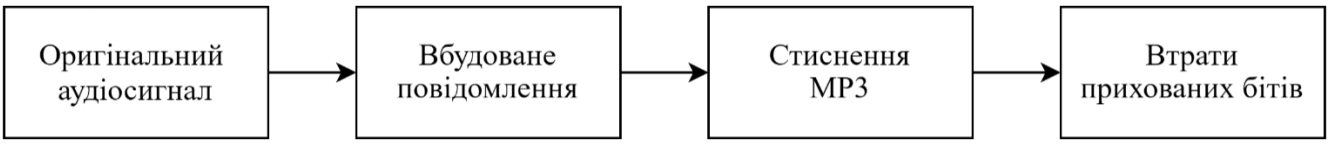


Рисунок 1.2 – Схематичне зображення впливу MP3-компресії на приховані дані

Вплив MP3-компресії залежить від того, у якій області (часовій або частотній) було здійснено вбудовування. Методи часової області (наприклад, LSB) особливо чутливі до будь-яких перетворень, оскільки вони безпосередньо змінюють семпли, що потім піддаються перерахунку при MDCT.

Методи частотної області демонструють кращу стійкість, оскільки деякі коефіцієнти частотного спектра зберігаються навіть після квантування. Це стосується передусім методів, які використовують QIM (Quantization Index Modulation) або spread-spectrum [2].

Для оцінки стійкості методів до стиснення використовуються такі метрики:

- BER (Bit Error Rate) — частка помилково витягнутих бітів;
- SNR (Signal-to-Noise Ratio) — співвідношення сигнал/шум;

– PESQ (Perceptual Evaluation of Speech Quality) — оцінка якості звучання на слух.

Таблиця 1.2 – Порівняння стійкості методів приховування до MP3-компресії

Метод	BER після MP3 (128 kbps)	SNR (дБ)	Примітка
LSB	0.45–0.70	25–30	Висока втрата даних
Echo Hiding	0.25–0.40	30–35	Збереження частини інформації
DCT-QIM	0.05–0.15	35–40	Висока стійкість
DWT-QIM	0.02–0.10	36–42	Дуже висока стійкість

Як видно з таблиці 1.2, методи частотної області, особливо ті, що базуються на QIM-квантуванні, показують значно меншу частоту помилок при витяганні даних після MP3-стиснення.

Основними чинниками, що знижують стійкість аудіостеганографічних методів до MP3-компресії є:

– квантування з втратами — дрібні варіації у значеннях амплітуди втрачаються;

– блокова обробка сигналу — відсутність синхронізації між вбудованими та реальними блоками;

– маскування частот — втрата високочастотних компонентів, які часто використовуються для вставки;

– варіативність параметрів компресії — залежно від бітрейту (96, 128, 192 kbps) ступінь втрат змінюється.

Для підвищення стійкості методів приховування до MP3-компресії дослідники пропонують такі рішення:

1. Використання QIM у частотній області — модифікація лише тих коефіцієнтів, які зберігаються після компресії [2].

2. Адаптивне вибирання піддіапазонів згідно з психоакустичною моделлю MP3 [3].

3. Додаткове шифрування ключ-блоку, який визначає порядок вставки, що ускладнює відновлення прихованих даних навіть у разі часткових втрат.

4. Використання надлишковості (error correction codes) для компенсації помилок при витяганні.

У результаті аналізу встановлено, що проблема стійкості аудіостеганографічних методів до MP3-компресії є критичною для практичного використання таких систем. Методи часової області (зокрема LSB) не забезпечують необхідної надійності, тоді як QIM-квантування у частотній області демонструє найкращі результати за показниками BER та SNR.
Тому подальші дослідження доцільно спрямувати на оптимізацію параметрів QIM-квантування з урахуванням психоакустичних властивостей MP3, а також використання шифрування ключ-блоку для додаткового захисту процесу вбудовування та витягання інформації.

1.3 Методи шифрування ключових блоків у стеганографічних системах

В Ключ є центральним елементом будь-якої стеганографічної системи, адже саме він визначає порядок, місце та параметри вбудовування даних у медіафайл. Втрата або розкриття ключа фактично означає повну компрометацію системи.

У класичних схемах ключ задає:

- послідовність позицій для вставки бітів;
- параметри модифікації амплітуд або коефіцієнтів;
- використання певної трансформації (DCT, DWT, MDCT тощо);
- синхронізацію між процесами вставки та витягання даних.

Тому для сучасних аудіостеганографічних систем виникає потреба не лише у стійкому до атак методі вбудовування (наприклад, QIM), але й у захищеному механізмі формування та зберігання ключа [7].

У науковій літературі розрізняють три основні типи ключів у стеганографії статичний, псевдовипадковий та блоковий (рисунок 1.3).

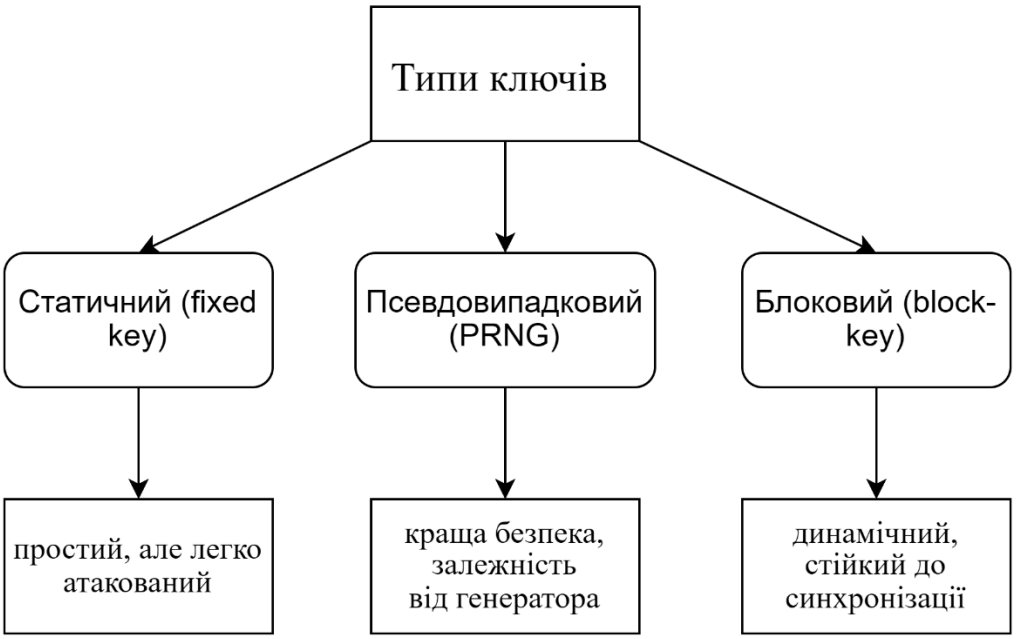


Рис. 1.3 – Основні типи ключів, що застосовуються у стеганографічних системах

Блокові ключі (block-key) мають перевагу, адже дозволяють формувати індивідуальні параметри вбудовування для кожного блоку сигналу. Це значно ускладнює виявлення закономірностей при стеганалізі. Такий підхід активно застосовується в сучасних гібридних системах приховування інформації в аудіо та відео.

Шифрування ключових блоків дозволяє підвищити рівень захисту системи, оскільки забезпечує приховування самої логіки вбудовування. Основні аспекти, які захищаються за рахунок цієї процедури:

- сам ключ від перехоплення чи аналізу;
- порядок вставки (який часто несе приховану структуру);
- кореляцію між блоками сигналу.

У результаті шифрування ключових блоків підвищується не лише стійкість системи до атак стеганалізу, але й її криптографічна надійність, оскільки навіть при частковому розкритті методики приховування зловмисник не зможе визначити, де саме і як розташовані приховані дані.

З практичної точки зору, шифрування ключ-блоку доцільно виконувати перед вбудовуванням у сигнал — тобто створювати попередньо зашифрований набір псевдовипадкових індексів, які потім використовуються QIM для модуляції частотних коефіцієнтів.

У загальному вигляді така схема може бути подана як (формула 1.3):

$$K' = E_{key}(K) \tag{1.3}$$

де

K— початковий ключ-блок,

E_{key}— функція шифрування за основним ключем системи,

K'— зашифрований ключ-блок, який використовується для генерації псевдовипадкової послідовності вставки.

Найчастіше використовуються методи шифрування наведенні в таблиці 1.3.

Таблиця 1.3 – Порівняльна характеристика методів шифрування ключ-блоків

Метод	Тип	Особливості	Стійкість до атак
AES (Advanced Encryption Standard)	Блочний симетричний	Висока швидкість, легка реалізація у Python/C++	Висока
RSA	Асиметричний	Висока безпека, але потребує більше ресурсів	Дуже висока
ChaCha20 / Salsa20	Потоковий	Підходить для потокових аудіосистем	Висока
Lightweight Cipher (Speck, Simon)	Блочний, спрощений	Оптимізований для IoT або low-power	Середня
Custom Permutation Cipher	Перестановка блоків	Додатковий рівень обфускації	Середня–висока

– Alanzy et al. (2023) запропонували комбінацію LSB та гібридного шифрування (AES + RSA), що дозволяє приховати дані з подвійним рівнем захисту. Проте через використання LSB метод не демонструє стійкості до втратних перетворень.

– Daiyrbayeva et al. (2025) використали адаптивне розміщення даних у зображеннях із попереднім шифруванням ключа симетричним методом, що підвищило стійкість до атак статистичного аналізу.

– Abdallah & Meshoul (2023) реалізували багатошаровий підхід до захисту аудіосигналів, де кожен рівень має власний ключ. Це підвищує безпеку, однак не враховує особливості MP3-компресії.

Використання шифрованого ключ-блоку у поєднанні з QIM-квантуванням у частотній області дає можливість побудувати адаптивну і захищену схему стеганографії, де:

1. Ключ шифрується за допомогою AES або ChaCha20;

2. Отриманий криптоблок генерує псевдовипадкову послідовність позицій у частотній області;

3. Вбудовування даних виконується через QIM-квантування обраних коефіцієнтів MDCT;

4. Декодування можливе лише при наявності правильного ключа та параметрів генератора.

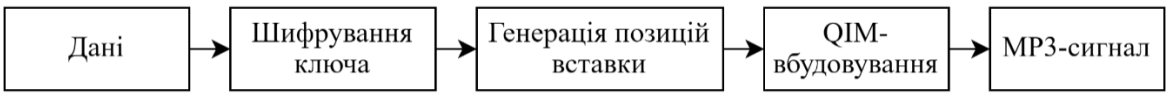


Рисунок 1.4 – Узагальнена схема комбінованого методу QIM + шифрування ключ-блоку

Такий підхід ускладнює стеганаліз навіть у разі часткових втрат, оскільки відновити порядок вбудовування без ключа практично неможливо. Це підтверджують результати сучасних досліджень [8].

У результаті аналізу встановлено, що застосування шифрування ключових блоків є ефективним способом підвищення безпеки стеганографічних систем.

Найбільш доцільним для реалізації у задачі підвищення стійкості приховування даних в аудіосигналах до стиснення MP3 є використання блочних шифрів (AES або ChaCha20) у комбінації з QIM-квантуванням у частотній області. Це дозволяє:

- захистити структуру та параметри вбудовування від атак;
- зменшити кореляцію між блоками;
- підвищити стійкість до компресії та спотворень сигналу;
- забезпечити адаптивність системи до різних форматів аудіо.

Отже, інтеграція шифрування ключ-блоку у QIM-схему є доцільним напрямом подальших досліджень та практичної реалізації у магістерській роботі.

1.4 Стійкість сучасних методів до компресії та атак стеганалізу

У сучасних системах аудіостеганографії велике значення надається адаптивності методу до конкретних характеристик сигналу. Адаптивні методи визначають місця вбудовування даних залежно від локальних властивостей аудіосигналу. Це дозволяє підвищити стійкість до компресії та атак стеганалізу. Основні підходи включають:

Локальна енергія сигналу. Ділянки сигналу з високою енергією менш чутливі до спотворень, що дозволяє надійно вбудовувати дані без значного погіршення якості звучання.

Психоакустичне маскування. Використання областей частот, менш сприйнятних для слуху, дозволяє збільшити обсяг вбудованих даних без помітного впливу на якість аудіо.

Динамічне квантування. Коефіцієнти QIM підлаштовуються під локальні властивості сигналу, що дозволяє знизити частку помилок (BER) після втратного стиснення.

Методи приховування інформації в аудіосигналах поділяються на дві основні категорії: методи часової області та методи частотної області. Вибір підходу безпосередньо впливає на стійкість, прозорість і надійність стеганографічної системи.

Найвідоміших методів часової області належать Least Significant Bit (LSB) та Phase Coding. Їх суть полягає у безпосередній модифікації амплітуди аудіосигналу або його фази. Наприклад, у методі LSB найменш значущі біти вибірок аудіо замінюються на біти прихованого повідомлення, що дозволяє забезпечити просту та швидку реалізацію.

Переваги:

– проста реалізація, не потребує складних перетворень сигналу;

– низька обчислювальна складність, що дозволяє застосовувати метод у реальному часі.

Недоліки:

– низька стійкість до втратного стиснення (MP3, AAC) та перетворень формату;

– висока чутливість до фільтрації, нормалізації гучності та шумових спотворень;

– низька безпека при аналізі сигналу, оскільки закономірності в LSB можуть бути легко виявлені.

За експериментальними даними [8], після перекодування сигналу у формат MP3 (128 kbps) втрачається до 80–90% прихованої інформації. Це робить LSB та подібні методи непридатними для застосування у випадках, коли аудіосигнал піддається компресії чи редагуванню.

Методи частотної групи (зокрема DCT, DWT, MDCT, QIM) базуються на вбудовуванні інформації у спектральні коефіцієнти сигналу після його перетворення з часової у частотну область. Такий підхід дозволяє приховувати дані у компонентах, менш помітних для людського слуху, завдяки врахуванню перцептивних характеристик аудіосприйняття.

Переваги:

– висока непомітність модифікацій — слухач не відрізняє оригінал від модифікованого сигналу;

– стійкість до втратного кодування (MP3, AAC), шумових атак і перетворень;

– можливість адаптивного вбудовування на основі психоакустичних моделей.

Недоліки:

– підвищена обчислювальна складність, необхідність виконання перетворень (DCT, DWT, MDCT);

– складність точного відновлення даних без втрат при надмірних модифікаціях або додатковій обробці сигналу.

Таким чином, методи частотної області є більш перспективними для практичного застосування, особливо у випадках, коли аудіо піддається компресії. Саме до цієї категорії належить метод QIM (Quantization Index Modulation), який використовується в даному дослідженні для підвищення стійкості приховання даних до MP3-компресії.

Для оцінки стійкості використовується ряд критеріїв наведених у таблиці 1.6.

Таблиця 1.6 – Порівняльна стійкість методів до MP3-компресії та атак стеганалізу

Метод	MP3 128 kbps	MP3 64 kbps	Виявлення стеганалізом	Примітки
LSB	15–20%	5–10%	Висока	Часова область
Phase Coding	50–60%	30–40%	Середня	Низька ємність
DCT	80–85%	70–75%	Низька	Частотна область
DWT	85–90%	75–80%	Низька	Частотна область
QIM	90–95%	85–90%	Дуже низька	Адаптивне квантування

Сучасні методи стеганалізу використовують статистичні та спектральні підходи:

– RS-аналіз (Regular-Singular) – виявляє закономірності в бітових рівнях;

– CCA (Correlation Coefficient Analysis) – виявляє кореляцію між блоками;

– Wavelet-Based Analysis – використовує високочастотні коефіцієнти для виявлення аномалій.

Експерименти показують, що методи LSB і Phase Coding піддаються RS-аналізу майже на 80–90% сигналів. Методи частотної області QIM, DWT залишаються практично невиявленими при використанні сучасних атак, особливо якщо використовується шифрування ключ-блоку, яке змінює порядок вставки даних.

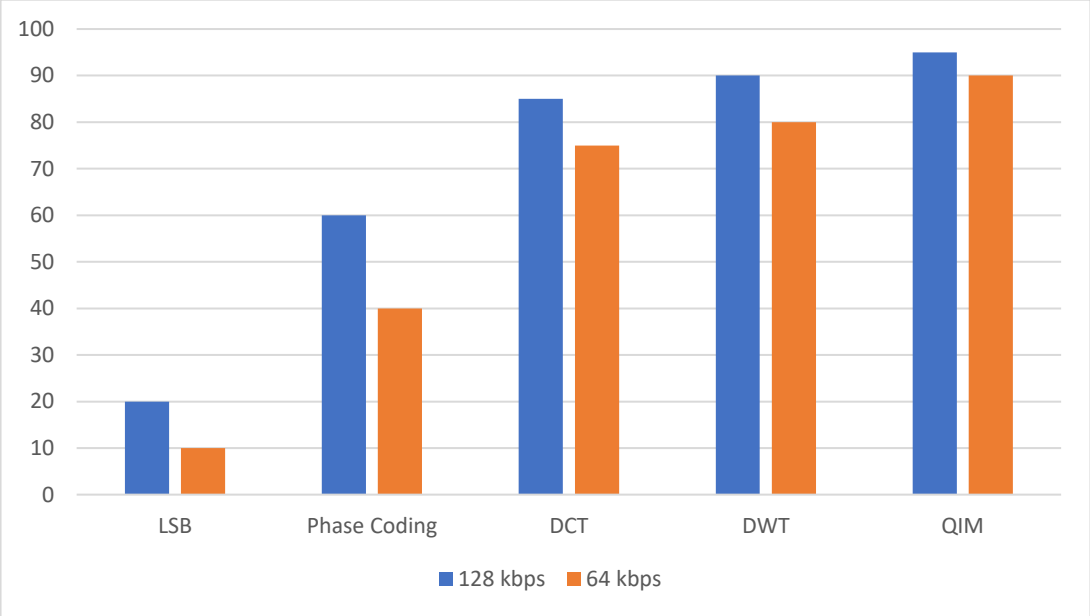


Рис. 1.6 – Відсоток втрати прихованих даних після MP3-компресії

Дана діаграма (рисунок 1.6) ілюструє, що QIM та DWT забезпечують найвищу стійкість до втратного кодування, тоді як часові методи практично повністю втрачають приховані дані при низьких бітрейтах.

Методи частотної області, особливо QIM, демонструють найвищу стійкість до компресії MP3 та атак стеганалізу. Часові методи LSB, Phase Coding є менш стійкими, тому їх доцільно використовувати тільки для прототипів або навчальних цілей. Додавання шифрування ключ-блоку суттєво підвищує захист від статистичних атак та ускладнює відновлення прихованих даних без наявності ключа.

1.5 Висновки до розділу

У першому розділі проведено аналітичне дослідження сучасних методів приховування інформації в аудіосигналах, здійснено класифікацію підходів за принципами вбудовування, областями дії, рівнем стійкості до атак і ступенем помітності спотворень.

Проведений аналіз дозволив виявити тенденції розвитку, обмеження класичних методів і визначити напрям подальшого вдосконалення технологій стеганографії для аудіо.

Аудіостеганографія як наукова дисципліна пройшла еволюцію від простих методів заміни наймолодших бітів (LSB) до складних гібридних систем, що поєднують цифрові перетворення сигналу, адаптивне квантування та криптографічний захист ключів.

Методи в часовій області (LSB, Phase Coding) характеризуються низькою обчислювальною складністю, але слабкою стійкістю до втратного стиснення, зокрема формату MP3, де частина інформації безповоротно втрачається.

Методи у частотній області (DCT, DWT, MDCT, QIM) демонструють вищу надійність, оскільки маніпулюють енергією сигналу на рівні спектральних коефіцієнтів, що узгоджується з властивостями людського слуху.

Найперспективнішими виявилися QIM (Quantization Index Modulation) та Spread Spectrum — вони забезпечують високу стійкість до атак і компресії, зберігаючи при цьому непомітність модифікацій.

Проведений порівняльний аналіз сучасних наукових джерел показав, що попри значний прогрес у галузі аудіостеганографії, існує низка невирішених проблем, серед яких втрата прихованих даних після MP3-кодування, що обмежує практичне використання методів у мультимедіа-комунікаціях, відсутність адаптивності до перцептивних особливостей сигналу — більшість підходів не враховують модель слухового сприйняття (HAS), високий ризик виявлення при статистичному аналізі стегосигналів (особливо у простих методах LSB), обмежена ємність каналів вбудовування без значного погіршення якості звуку.

Водночас останні дослідження, зокрема H. Abdallah і S. Meshoul, показали, що багатошарове шифрування аудіосигналів у частотній області істотно підвищує рівень захищеності, тоді як E. Daiyrbayeva у своїй роботі довела ефективність адаптивного реверсивного вбудовування на основі оцінки локальної активності. Результати Alanzy et al. підтвердили доцільність комбінування стеганографії з криптографією для формування єдиного багаторівневого захисту.

На основі проведеного аналізу можна зробити висновок, що для забезпечення стійкості прихованих даних до MP3-стиснення доцільно використовувати квантування індексів (QIM) у частотній області. Використання MDCT (Modified Discrete Cosine Transform) як базової трансформації дозволяє здійснювати вбудовування в області, безпосередньо пов’язаній з психоакустичною моделлю MP3.

Додаткове шифрування ключ-блоку дає змогу гарантувати криптографічну безпеку каналу стеганопередачі, навіть у разі часткового відновлення або розкриття структури вбудовування.

Таким чином, гібридний метод “QIM у частотній області та шифрування ключ-блоку” є логічно обґрунтованим вибором для подальшої розробки.

Проведено класифікацію методів аудіостеганографії та визначено їх переваги й недоліки. Встановлено, що класичні методи LSB, Phase Coding не забезпечують достатньої стійкості до компресії, тоді як частотні QIM, DWT, MDCT дозволяють зберегти приховану інформацію після MP3-кодування. Визначено, що застосування криптографічних механізмів на рівні ключів дозволяє значно підвищити безпечність системи. Обґрунтовано необхідність створення нового гібридного методу, який забезпечуватиме підвищену стійкість до втратного кодування та захищеність ключової інформації.

Таким чином, у результаті аналітичного дослідження було визначено актуальність теми магістерської кваліфікаційної роботи та сформовано теоретичну основу для розроблення моделі методу приховування даних в аудіосигналах на основі QIM-квантування у частотній області та шифрування ключ-блоку, що буде реалізовано у другому розділі.

2 РОЗРОБЛЕННЯ МЕТОДУ ПІДВИЩЕННЯ СТІЙКОСТІ ПРИХОВУВАННЯ ДАНИХ В АУДІОСИГНАЛАХ ДО СТИСНЕННЯ MP3 НА ОСНОВІ QIM-КВАНТУВАННЯ ТА ШИФРУВАННЯ КЛЮЧ-БЛОКУ

2.1 Розроблення методу підвищення стійкості приховування даних в аудіосигналах

Розроблення ефективного методу стеганографічного приховування інформації в аудіосигналах, стійкого до стиснення MP3, вимагає комплексного підходу, що поєднує сучасні досягнення цифрової обробки сигналів, криптографії та теорії інформації. Базовою ідеєю запропонованого методу є використання квантування з індексною модуляцією (Quantization Index Modulation, QIM) у поєднанні з криптографічним захистом ключових параметрів вбудовування.

Традиційні стеганографічні методи часто демонструють недостатню стійкість до стиснення з втратами, оскільки алгоритми стиснення, зокрема MP3, суттєво модифікують частотні характеристики аудіосигналу [10]. Психоакустична модель, що лежить в основі MP3, агресивно усуває компоненти сигналу, які вважаються непомітними для людського слуху, що призводить до втрати прихованих даних.

Метод QIM-квантування, вперше детально описаний у роботах Chen і Wornell, базується на принципі модифікації квантованих значень аудіосигналу відповідно до біту повідомлення, що вбудовується. На відміну від простої заміни LSB (найменш значущих бітів), QIM забезпечує кращу стійкість до шумів та спотворень завдяки використанню множини квантувальників.

Ключовою особливістю запропонованого методу є додатковий рівень захисту через криптографічне шифрування параметрів вбудовування, що формують так званий "ключ-блок". Цей підхід забезпечує не лише стійкість до стиснення, але й криптографічну безпеку, ускладнюючи виявлення та вилучення прихованої інформації без знання секретного ключа [11].

Структурна модель запропонованого методу підвищення стійкості приховування даних в аудіосигналах складається з чотирьох основних функціональних блоків (рисунок 2.1).

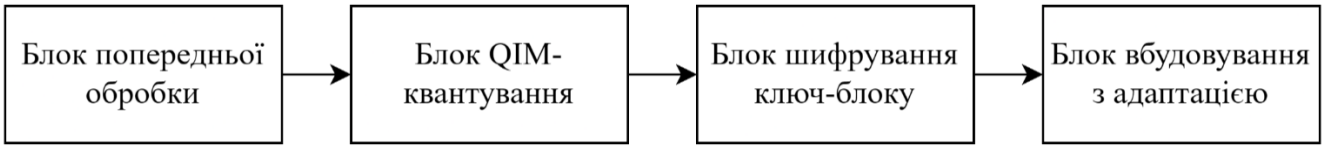


Рисунок 2.1 – Структурна модель методу підвищення стійкості приховування даних в аудіосигналах

Блок попередньої обробки відповідає за підготовку аудіоконтейнера до вбудовування інформації. Цей блок виконує наступні функції:

- сегментацію аудіосигналу на фрейми фіксованої довжини;
- застосування віконної функції для зменшення крайових ефектів;
- перетворення у частотну область за допомогою модифікованого дискретного косинусного перетворення (MDCT);
- аналіз психоакустичних характеристик кожного фрейму.

Використання MDCT обумовлено тим, що саме це перетворення застосовується в кодеках MP3 та AAC [12]. Робота у тій же частотній області, що й алгоритм стиснення, дозволяє краще передбачити поведінку вбудованих даних після компресії.

Блок QIM-квантування реалізує основний механізм вбудовування даних. Для кожного біту повідомлення $m_i \in \{0,1\}$обирається пара квантувальників Q_0та Q_1з кроком квантування Δ (формула 2.1).

$$Q_b(x) = \Delta \cdot \left\lfloor \frac{x}{\Delta} + \frac{1}{2} \right\rfloor + \frac{b \cdot \Delta}{2}, b \in \{0,1\} \quad (2.1)$$

де x– коефіцієнт MDCT, який модифікується, Δ– крок квантування, b– біт повідомлення.

Вибір кроку квантування Δ є критичним параметром, що визначає компроміс між непомітністю та стійкістю. Надто малий крок забезпечує високу непомітність, але знижує стійкість до шумів та стиснення. Надто великий крок підвищує стійкість, але може призвести до помітних артефактів.

Блок шифрування ключ-блоку забезпечує криптографічний захист параметрів вбудовування. Ключ-блок містить наступну інформацію:

- індекси коефіцієнтів MDCT, обраних для модифікації;
- значення кроків квантування для кожної позиції;
- параметри адаптації до психоакустичної моделі;
- контрольні суми для перевірки цілісності [13].

Для шифрування ключ-блоку використовується алгоритм AES-256 у режимі CBC (Cipher Block Chaining) [14]. Вибір AES обумовлений його високою криптографічною стійкістю та ефективністю реалізації на сучасних процесорах.

Блок вбудовування з адаптацією виконує фінальну інтеграцію модифікованих коефіцієнтів у аудіосигнал з урахуванням психоакустичних характеристик. Адаптивний механізм коригує силу вбудовування відповідно до маскувальних властивостей кожного фрейму, забезпечуючи оптимальний баланс між непомітністю та робастністю.

Архітектура запропонованого методу організована у вигляді багаторівневої системи з чіткою ієрархією обробки даних (рисунок 2.2).

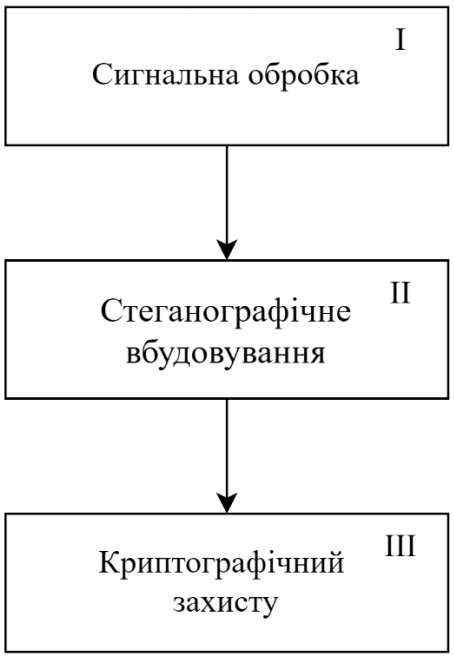


Рисунок 2.2 – Архітектура методу підвищення стійкості приховування даних в аудіосигналах

Рівень сигнальної обробки відповідає за перетворення аудіосигналу з часової області у частотну та назад. На цьому рівні реалізуються наступні компоненти:

Модуль частотного аналізу виконує пряме MDCT перетворення з розміром вікна 1152 відліки та перекриттям 50% (формула 2.2).

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot w(n) \cdot \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2} + \frac{N}{2}\right)\left(k + \frac{1}{2}\right)\right] \quad (2.2)$$

де x(n)– вхідний аудіосигнал, w(n)– віконна функція (зазвичай вікно Кайзера-Бесселя), N– розмір вікна, k– індекс частотного коефіцієнта[15].

Модуль психоакустичного аналізу обчислює порогові значення маскування для кожної частотної смуги на основі моделі MP3. Це дозволяє визначити, які коефіцієнти можна модифікувати з мінімальним впливом на прийнятну якість (формула 2.3).

$$T_m(k) = 10^{\frac{T_q(k)+SNR_{target}}{10}} \quad (2.3)$$

де $T_m(k)$– поріг маскування для коефіцієнта k, $T_q(k)$– поріг тихого звучання, SNR_{target}– цільове співвідношення сигнал-шум (зазвичай 20-25 дБ).

Рівень стеганографічного вбудовування реалізує власне процес модифікації коефіцієнтів MDCT згідно з QIM-квантуванням. Ключовими компонентами цього рівня є:

1. Модуль вибору коефіцієнтів визначає оптимальні позиції для вбудовування даних на основі критеріїв стійкості та непомітності. Використовується евристичний алгоритм, що враховує:

- величину коефіцієнта (переважно середньочастотні компоненти);
- відношення до порогу маскування;
- стабільність коефіцієнта при стисненні;
- спектральну плоскість навколишніх компонент.

2. Модуль QIM-модифікації застосовує квантування відповідно до біту повідомлення та параметрів ключ-блоку. Для підвищення стійкості

використовується розширений варіант QIM з випадковим дизерингом (формула 2.4).

$$x'(k) = Q_{m_i}\big(x(k) + d(k)\big) - d(k) \quad (2.4)$$

де $d(k)$– псевдовипадкове значення дизерингу, що генерується на основі секретного ключа.

3. Модуль контролю якості перевіряє, чи модифікований коефіцієнт задовольняє обмеженням непомітності. Якщо модифікація призводить до перевищення порогу маскування, виконується перерозподіл біту на альтернативну позицію [16].

Рівень криптографічного захисту забезпечує конфіденційність та автентичність параметрів вбудовування. Він містить:

Модуль генерації ключ-блоку формує структуровану послідовність параметрів, що включає індекси позицій, кроки квантування та контрольні суми. Структура ключ-блоку організована у форматі TLV (Type-Length-Value) для гнучкості та розширюваності [17].

Модуль шифрування застосовує AES-256-CBC до ключ-блоку з використанням секретного ключа та вектора ініціалізації (IV), що генерується криптографічно стійким генератором псевдовипадкових чисел (формула 2.5).

$$C = \text{AES-256-CBC}_K(KB, IV) \quad (2.5)$$

де C– зашифрований ключ-блок, КВ– відкритий ключ-блок, К– секретний ключ, IV– вектор ініціалізації.

Модуль управління ключами реалізує протокол безпечного зберігання та обміну криптографічними ключами. Використовується схема на основі функції виведення ключа PBKDF2 (Password-Based Key Derivation Function 2) для генерації робочих ключів з парольної фрази (формула 2.6).

$$K = \text{PBKDF2}(\text{password},\text{salt},\text{iterations},\text{keylen}) \quad (2.6)$$

де параметр iterations встановлюється не менше 100000 для захисту від атак перебору. Для наочного представлення послідовності операцій розроблено блок-

схему методу (рисунок 2.3), яка відображає основні етапи вбудовування інформації та взаємозв'язки між функціональними блоками.

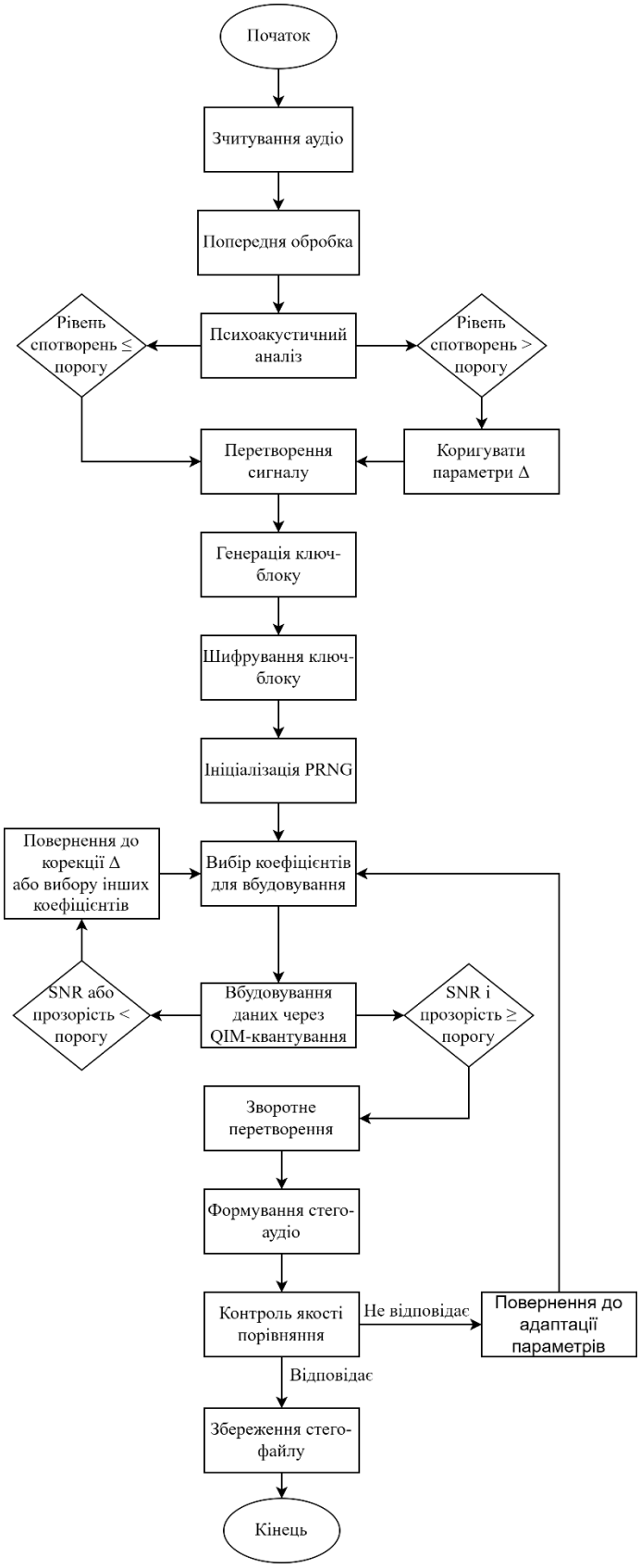


Рисунок 2.3 – Блок-схема роботи методу

Блок-схема демонструє розгалужену структуру алгоритму з численними точками прийняття рішень, що забезпечують адаптивність методу до характеристик конкретного аудіосигналу. Ключовими елементами блок-схеми є:

– блоки перевірки умов, що контролюють дотримання обмежень на непомітність та ємність;

– ітераційні цикли обробки фреймів та позицій вбудовування;

– розгалуження для вибору альтернативних стратегій при виникненні конфліктів;

– підпрограми психоакустичного аналізу та криптографічних операцій [18].

Особливістю блок-схеми є наявність зворотних зв'язків, що дозволяють коригувати параметри вбудовування у разі виявлення проблем на етапі контролю якості. Це забезпечує робастність методу до різноманітних типів аудіоконтенту.

Стійкість запропонованого методу до стиснення MP3 може бути проаналізована через призму теорії інформації та обробки сигналів. Ймовірність помилки декодування біту після стиснення визначається співвідношенням (формула 2.7).

$$P_e = Q\left(\frac{\Delta - 2\sigma_n}{\sqrt{2}\sigma_n}\right) \tag{2.7}$$

де $Q(\cdot)$– функція помилок, σ_n– середньоквадратичне відхилення шуму, внесеного стисненням.

Для забезпечення прийнятної ймовірності помилки ($P_e < 10^{-3}$) необхідно, щоб крок квантування задовольняв умові (формула 2.8).

$$\Delta \geq 2\sigma_n + \sqrt{2}\sigma_n \cdot Q^{-1}(P_e) \approx 2\sigma_n(1 + 2.4) = 6.8\sigma_n \tag{2.8}$$

Експериментальні дослідження показують, що для типових аудіосигналів при стисненні MP3 з бітрейтом 128 кбіт/с величина σ_nскладає приблизно 0.01-0.05 від динамічного діапазону коефіцієнтів MDCT [19]. Отже, мінімальний крок квантування повинен бути порядку 0.07-0.34 від типової амплітуди коефіцієнтів у середньочастотному діапазоні.

Крім того, застосування криптографічного дизерингу забезпечує рівномірний розподіл помилок квантування, що підвищує стійкість до статистичних атак

стегоаналізу. Ентропія прихованого повідомлення після дизерингу наближається до максимальної (формула 2.9).

$$H(M \mid C) \approx \log_2(2) = 1 \text{ біт} \tag{2.9}$$

де M– повідомлення, C– стего-контейнер.

Обчислювальна складність запропонованого методу є важливою характеристикою, що визначає його практичну застосовність. Аналіз складності виконується окремо для основних етапів алгоритму [20].

MDCT перетворення: Складність обчислення MDCT для сигналу довжиною Nвідліків з використанням швидких алгоритмів становить $O(N\log\ N)$операцій. Для стандартного розміру вікна 1152 відліки це приблизно 11000 операцій на фрейм.

Психоакустичний аналіз: Складність обчислення порогів маскування залежить від деталізації моделі. Спрощена модель вимагає $O(N)$операцій, повна модель MP3 – близько $O(N\log\ N)$.

QIM-квантування: Для вбудовування Lбіт необхідно виконати Lоперацій квантування, кожна з яких має складність $O(1)$. Загальна складність: $O(L)$.

Шифрування AES: Складність шифрування ключ-блоку розміром Kбайт за допомогою AES-256 становить $O(K)$операцій блочного шифрування. Оскільки розмір ключ-блоку пропорційний кількості вбудованих бітів, $K \approx c \cdot L$, де c– константа (зазвичай 8-16 байт на біт).

Загальна обчислювальна складність методу для аудіосигналу тривалістю Tсекунд з частотою дискретизації f_sГц (формула 2.10).

$$\text{Complexity} = O\left(\frac{T \cdot f_s}{F} \cdot F \log F\right) + O(L) + O(c \cdot L) \tag{2.10}$$

де F– розмір фрейму (1152 відліки).

Спрощуючи: $\text{Complexity} = O(T \cdot f_s \log\ F) + O(L)$.

Для типового випадку (44.1 кГц, 3 хвилини аудіо, 10000 біт повідомлення) це становить приблизно 8×10^9операцій, що може бути виконано на сучасному процесорі за декілька секунд [21].

Ефективність запропонованого методу суттєво залежить від коректного вибору параметрів налаштування. Основні параметри та рекомендовані значення наведені у таблиці 2.1.

Таблиця 2.1 – Параметри налаштування методу

Параметр	Позначення	Рекомендоване значення	Вплив на характеристики
Базовий крок квантування	Δ_0	0.1-0.3	Баланс стійкість/непомітність
Розмір фрейму	F	1152 відліки	Сумісність з MP3
Перекриття фреймів	O	50%	Зменшення артефактів
Діапазон частот вбудовування	$[f_{\min}, f_{\max}]$	[1-8] кГц	Оптимальна стійкість
Цільове SNR	SNR_{target}	20-25 дБ	Якість стего-сигналу
Кількість ітерацій PBKDF2	N_{iter}	100000	Криптографічна стійкість
Коефіцієнт адаптації	α	0.5-2.0	Динамічна корекція Δ

Базовий крок квантування Δ_0 є найкритичнішим параметром методу. Занадто малі значення ($\Delta_0 < 0.05$) призводять до високої чутливості до шуму стиснення, що збільшує ймовірність помилок декодування до неприйнятних рівнів ($P_e > 0.1$). Надмірно великі значення ($\Delta_0 > 0.5$) можуть спричинити помітні спотворення аудіосигналу, особливо в тихих фрагментах.

Діапазон частот вбудовування обирається на основі компромісу між стійкістю до стиснення та психоакустичною непомітністю. Низькочастотні компоненти (< 1 кГц) характеризуються високою енергією, але їх модифікація легше виявляється людським слухом. Високочастотні компоненти (> 8 кГц) добре

маскуються, але часто повністю усуваються при стисненні з низьким бітрейтом. Середньочастотний діапазон 1-8 кГц забезпечує оптимальний баланс.

Коефіцієнт адаптації αвизначає ступінь динамічної корекції кроку квантування відповідно до локальних характеристик сигналу. Адаптивний крок квантування обчислюється як наведено у формулі 2.11.

$$\Delta_{\text{adapt}} = \Delta_0 \cdot \alpha \cdot \left(\frac{| X(k) |}{T_m(k)} \right)^{\beta} \tag{2.11}$$

де β– експонента адаптації (зазвичай 0.3-0.5), $X(k)$– коефіцієнт MDCT, $T_m(k)$– поріг маскування.

Для забезпечення максимальної сумісності з екосистемою MP3, метод розроблено з урахуванням специфіки внутрішньої структури цього формату. MP3 використовує гібридну схему кодування, що поєднує фільтрацію поліфазним банком фільтрів та MDCT перетворення.

Стандартний кодер MP3 (згідно ISO/IEC 11172-3) розділяє аудіосигнал на 32 субсмуги за допомогою поліфазного аналізу, після чого кожна субсмуга додатково обробляється MDCT. Для спрощення реалізації запропонований метод працює безпосередньо з MDCT коефіцієнтами, оминаючи етап поліфазної фільтрації.

Така архітектура має дві переваги:

1. Незалежність від специфічної реалізації кодера – метод сумісний з будь-яким MP3-кодером, що дотримується стандарту;

2. Можливість прямої модифікації MP3-файлів – при наявності доступу до декодованих MDCT коефіцієнтів можна вбудовувати інформацію безпосередньо у стиснуті файли без повного декодування.

Хоча QIM-квантування забезпечує базову стійкість до шуму стиснення, додатковий механізм виявлення та корекції помилок значно підвищує надійність методу. У запропонованому підході використовується каскадна схема захисту:

Рівень 1: Коди Ріда-Соломона застосовуються до повідомлення перед вбудовуванням. RS-коди здатні виправляти пакетні помилки, що часто виникають при стисненні через локальні спотворення у певних частотних смугах (формула 2.12).

$$\text{RS}(n, k, t): n = k + 2t \tag{2.12}$$

де n– довжина кодового слова, k– довжина інформаційних символів, t– кількість виправляємих помилок.

Для типового застосування використовується RS(255, 223, 16), що дозволяє виправити до 16 пошкоджених байтів у блоці 255 байт.

Рівень 2: Контрольні суми CRC вбудовуються для верифікації цілісності окремих сегментів повідомлення. Це дозволяє виявити залишкові помилки, які не були скориговані RS-кодом (формула 2.13).

$$\text{CRC-32}(M) = M(x) \cdot x^{32} \text{mod } G(x) \tag{2.13}$$

де $G(x) = x^{32} + x^{26} + x^{23} + \ldots + x + 1$– генераторний поліном.

Рівень 3: Повторне вбудовування критичних біт застосовується для найважливіших частин повідомлення (заголовки, синхронізаційні маркери). Кожен критичний біт вбудовується у кілька незалежних позицій, а при декодуванні використовується мажоритарне голосування (формула 2.14).

$$\hat{m}_i = \text{majority}\left(m_i^{(1)}, m_i^{(2)}, \ldots, m_i^{(r)}\right) \tag{2.14}$$

де r– коефіцієнт повторення (зазвичай 3 або 5).

Для оцінки переваг запропонованого методу проведено порівняльний аналіз з існуючими підходами до стеганографії в аудіосигналах (таблиця 2.2).

Таблиця 2.2 – Порівняльна характеристика методів аудіостеганографії

Метод	Ємність (біт/с)	Стійкість до MP3	SNR (дБ)	Складність
LSB	100-500	Низька	40-50	Низька
Phase Coding	10-50	Середня	30-40	Середня
Echo Hiding	50-200	Середня	25-35	Середня
Spread Spectrum	20-100	Висока	20-30	Висока
Базовий QIM	50-150	Середня	25-35	Висока
Запропонований метод	80-250	Висока	20-30	Висока

Як видно з таблиці, запропонований метод забезпечує конкурентну ємність при одночасно високій стійкості до стиснення MP3. Основна перевага полягає у поєднанні QIM-квантування з адаптивним вибором параметрів та криптографічним захистом, що недоступно в базових реалізаціях.

Метод LSB, хоча й забезпечує високу ємність та SNR, практично непридатний для застосувань, що вимагають стійкості до стиснення, оскільки найменш значущі біти повністю знищуються при MP3-кодуванні.

Методи на основі розширеного спектру (Spread Spectrum) демонструють високу стійкість, але мають обмежену ємність через необхідність розподілу кожного біту по великій кількості коефіцієнтів.

2.2 Цілі та вимоги до процесу підвищення стійкості приховування даних в аудіосигналах

Основною метою розробленого методу є створення ефективної системи стеганографічного приховування інформації в аудіосигналах, яка забезпечує стійкість до стиснення з втратами, зокрема MP3, при збереженні високої якості аудіоконтейнера та криптографічної безпеки прихованих даних. Ця загальна мета конкретизується через ряд специфічних цілей.

Ціль 1: забезпечення стійкості до стиснення MP3 полягає у розробці механізмів вбудовування, які зберігають цілісність прихованої інформації після застосування алгоритмів стиснення з втратами. Статистичні дослідження показують, що традиційні методи LSB втрачають до 95% вбудованих даних при стисненні MP3 з бітрейтом 128 кбіт/с. Запропонований метод повинен забезпечити коефіцієнт витягування (extraction rate) не менше 98% при бітрейті 128 кбіт/с та не менше 95% при бітрейті 64 кбіт/с.

Математично це формулюється як вимога до ймовірності помилки декодування біту (формула 2.15).

$$P_e^{\text{MP3}} < 0.02 \text{для бітрейту } 128 \text{ кбіт/с} \qquad (2.15)$$

де P_e^{MP3}– ймовірність помилкового декодування біту після стиснення та декомпресії MP3.

Ціль 2: Досягнення високої непомітності модифікацій передбачає мінімізацію перцептивних спотворень аудіосигналу, викликаних вбудовуванням інформації. Непомітність оцінюється як об'єктивними метриками (SNR, ODG), так і суб'єктивними тестами з залученням експертів-слухачів.

Цільові показники якості для стего-аудіо встановлюються на рівні:

- Signal-to-Noise Ratio (SNR): SNR $\geq$ 25дБ;
- Objective Difference Grade (ODG): ODG ≥ -1.0(практично непомітна різниця);
- Mean Opinion Score (MOS): MOS $\geq$ 4.0з 5.0 (добра якість).

Ціль 3: Оптимізація інформаційної ємності спрямована на максимізацію кількості даних, що можуть бути приховані в аудіоконтейнері заданої тривалості при дотриманні обмежень на якість та стійкість. Для практичних застосувань мінімальна ємність повинна становити 50-100 біт на секунду аудіо якості CD (44.1 кГц, 16 біт, стерео). Теоретична верхня межа ємності визначається теоремою пропускної здатності каналу з шумом Шеннона (формула 2.16).

$$C = B\log_2(1 + \frac{S}{N}) \quad (2.16)$$

де C– ємність (біт/с), B– ефективна смуга пропускання, S/N– відношення сигнал-шум.

Для типового аудіосигналу з ефективною смугою 8 кГц та SNR 25 дБ ($S/N \approx 316$), теоретична ємність становить приблизно 66 кбіт/с. Практична реалізація досягає 10-15% від теоретичної межі, що дає цільову ємність 80-250 біт/с.

Ціль 4: Забезпечення криптографічної безпеки передбачає захист від несанкціонованого виявлення, вилучення та модифікації прихованої інформації. Метод повинен протистояти як статистичним атакам стегоаналізу, так і спробам криптоаналізу параметрів вбудовування.

Мінімальний рівень криптографічної стійкості встановлюється еквівалентним AES-256, що відповідає стійкості до атаки перебору порядку 2^{256}операцій. Додатково, система повинна забезпечувати семантичну безпеку (semantic security) – неможливість отримання інформації про повідомлення без

ключа, стійкість до атак на основі вибраного відкритого тексту (chosen-plaintext attack), захист від атак на основі вибраного стего-тексту (chosen-stego attack).

Ціль 5: Досягнення обчислювальної ефективності спрямована на створення методу, придатного для практичного використання на сучасному апаратному забезпеченні. Час обробки аудіо повинен бути порівнянним або кращим за час природного відтворення (формула 2.17).

$$T_{\text{processing}} \leq k \cdot T_{\text{audio}} \tag{2.17}$$

де $T_{\text{processing}}$– час обробки, T_{audio}– тривалість аудіо, k– коефіцієнт реального часу (цільове значення $k \leq 10$).

Система повинна автоматично визначати оптимальні параметри на основі характеристик аудіосигналу крок квантування Δ для кожної позиції вбудовування, розподіл бітів повідомлення між частотними смугами, вибір стратегії вбудовування (агресивна/консервативна) залежно від специфіки контенту.

Адаптивність критично важлива для роботи з різноманітним аудіоконтентом – від класичної музики з широким динамічним діапазоном до сильно стиснутої популярної музики. Математична модель адаптації базується на локальних характеристиках сигналу (формула 2.18).

$$\Delta_i = \Delta_0 \cdot f(E_i, T_i, S_i) \tag{2.18}$$

де E_i– локальна енергія сигналу, T_i– поріг маскування, S_i– спектральна плоскість, $f(\cdot)$– адаптивна функція.

Реалізація багаторівневої системи захисту від помилок коди Ріда-Соломона RS(255, 223, 16) для корекції пакетних помилок, CRC-32 для виявлення залишкових помилок, повторне вбудовування критичних бітів (заголовків, синхромаркерів). Ефективність корекції помилок оцінюється через коефіцієнт виграшу кодування (coding gain):

$$G_c = \frac{\text{SNR}_{\text{coded}}}{\text{SNR}_{\text{uncoded}}}$$

Для RS(255, 223, 16) теоретичний виграш становить приблизно 3-5 дБ у каналах з пакетними помилками.

Система управління криптографічними ключами повинна підтримувати генерацію ключів на основі парольних фраз (PBKDF2), зберігання ключів у зашифрованому вигляді, експорт/імпорт ключів у стандартних форматах (PEM, DER), ієрархічну структуру ключів (master key → derived keys).

Схема виведення ключа реалізується згідно зі стандартом PKCS #5 v2.0:

$$DK = \text{PBKDF2}(\text{PRF},\text{Password},\text{Salt}, c, dkLen)$$

де PRF – псевдовипадкова функція (HMAC-SHA256), c– кількість ітерацій (≥100000), dkLen – довжина ключа (256 біт).

Якість стего-аудіо є критичним фактором успіху методу, оскільки помітні спотворення демаскують присутність прихованої інформації.

Згідно з рекомендаціями ITU-R BS.1116, суб'єктивна оцінка проводиться з використанням методу MUSHRA (MUltiple Stimuli with Hidden Reference and Anchor). Група з мінімум 20 експертів-слухачів оцінює якість стего-аудіо за шкалою від 0 до 100.

Інтерпретація результатів MUSHRA:

- 80-100: відмінна якість (excellent);
- 60-80: добра якість (good);
- 40-60: задовільна якість (fair);
- 20-40: погана якість (poor);
- 0-20: дуже погана якість (bad).

Цільовий показник: MUSHRA score ≥ 70 (добра якість).

Стійкість (robustness) визначає здатність методу зберігати приховані дані при різноманітних трансформаціях стего-сигналу. Основна вимога – збереження інформації при MP3-стисненні:

- бітрейт 320 кбіт/с: BER ≤ 0.001 (99.9% коректних біт);
- бітрейт 192 кбіт/с: BER ≤ 0.005 (99.5% коректних біт);
- бітрейт 128 кбіт/с: BER ≤ 0.02 (98% коректних біт);
- бітрейт 64 кбіт/с: BER ≤ 0.05 (95% коректних біт).

BER (Bit Error Rate) – відношення помилкових біт до загальної кількості біт.

Додатково, метод повинен демонструвати прийнятну стійкість до інших кодеків:

- AAC (iTunes): BER ≤ 0.025 при 128 кбіт/с;
- Vorbis (OGG): BER ≤ 0.03 при 128 кбіт/с;
- Opus: BER ≤ 0.04 при 128 кбіт/с.

Стійкість до зашумлення. Додавання адитивного білого гауссівського шуму (AWGN):

$$\tilde{x}(n) = x(n) + w(n), w(n) \sim \mathcal{N}(0, \sigma^2)$$

Вимога: BER ≤ 0.1 при SNR = 20 дБ після додавання шуму.

Зміна частоти дискретизації є поширеною операцією при конвертації між форматами:

- 44.1 кГц → 48 кГц → 44.1 кГц: BER ≤ 0.03;
- 44.1 кГц → 22.05 кГц → 44.1 кГц: BER ≤ 0.08.

Стійкість до фільтрації. Застосування фільтрів (lowpass, highpass, equalizer):

- фільтр нижніх частот (cutoff 16 кГц): BER ≤ 0.02;
- еквалайзер (±6 дБ у різних смугах): BER ≤ 0.05.

Стійкість до нелінійних спотворень, динамічна компресія (ratio 4:1, threshold -20 dB): BER ≤ 0.06, обрізання амплітуди (clipping 5%): BER ≤ 0.1, нормалізація гучності: BER ≤ 0.01.

Криптографічні аспекти методу повинні відповідати сучасним стандартам інформаційної безпеки. Простір ключів повинен бути достатньо великим для унеможливлення перебору розмір ключа: 256 біт (AES-256), складність перебору $2^{256} \approx 1.16 \times 10^{77}$операцій. При швидкості перебору 10^{18}ключів/секунду, наднереалістична оцінка з урахуванням квантових комп'ютерів майбутнього, час перебору становить 3.7×10^{51}років.

Стійкість до криптоаналітичних атак диференційний криптоаналіз: AES-256 стійкий до відомих атак, лінійний криптоаналіз: складність 2^{128}для AES-256, атаки на споріднені ключі (related-key attacks): стійкість через використання PBKDF2.

Приховані дані не повинні створювати статистично виявлювані аномалії:

1. Chi-square тест на рівномірність розподілу модифікованих коефіцієнтів (формула 2.19).

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \tag{2.19}$$

Вимога: $\chi^2 < \chi^2_{\text{crit}}(k-1, 0.05)$(гіпотеза про рівномірність не відхиляється).

2. Sample Pairs Analysis (SPA) – атака на LSB-стеганографію. Запропонований метод повинен бути стійким до SPA завдяки використанню QIM замість прямої заміни LSB.

3. Weighted Stego-image (WS) аналіз – перевірка відхилень статистики стего від очікуваної: Detection Rate ≤ 0.55, не краще випадкового вгадування.

Захист від side-channel атак timing attacks: використання constant-time криптографічних примітивів, power analysis: не застосовується для програмної реалізації, cache attacks: уникнення табличних реалізацій AES, використання AES-NI інструкцій. Для підтвердження переваг запропонованого методу необхідне порівняння з актуальними альтернативами.

Таблиця 2.3 – Порівняння вимог до різних методів аудіостеганографії

Характеристика	LSB	Phase Coding	Echo Hiding	Spread Spectrum	Базовий QIM	Запропонований метод
SNR (дБ)	40-50	30-40	25-35	20-30	25-35	25-30
BER після MP3 128	0.50+	0.15-0.25	0.10-0.20	0.03-0.08	0.08-0.15	≤0.02
Ємність (біт/с)	100-500	10-50	50-200	20-100	50-150	80-250
Crypto захист	Ні	Ні	Ні	Частково	Ні	Так (AES-256)
Адаптивність	Низька	Низька	Середня	Висока	Середня	Висока

Benchmark тестування проведення стандартизованих тестів на однаковому наборі даних:

- StirMark Benchmark – набір атак для оцінки стійкості водяних знаків
- BOSS (Break Our Steganographic System) – змагання зі стегоаналізу
- AudioSet – великий набір різноманітних аудіосемплів від Google

Сформульовані вимоги створюють чітку систему координат для розробки, тестування та оцінки запропонованого методу підвищення стійкості стеганографічного приховування даних в аудіосигналах.

2.3 Вибір мови програмування та середовища розробки

Вибір мови програмування є стратегічним рішенням, що суттєво впливає на всі аспекти розробки, від швидкості імплементації до фінальної продуктивності системи. Для реалізації методу підвищення стійкості приховування даних в аудіосигналах встановлено наступні критерії вибору.

Наявність бібліотек для обробки аудіо та сигналів. Реалізація MDCT перетворення, психоакустичних моделей та операцій з аудіофайлами потребує потужних бібліотек. Ідеальна мова повинна мати бібліотеки для читання/запису аудіоформатів (MP3), інструменти для частотного аналізу (FFT, MDCT, вейвлет-перетворення), засоби для обробки сигналів (фільтрація, ресемплінг, конвертація).

Продуктивність обчислень. Обробка аудіосигналів є обчислювально інтенсивною задачею. Мова повинна забезпечувати ефективну роботу з числовими масивами (векторизація), можливість використання низькорівневих оптимізацій, підтримку багатопотоковості та паралелізму, інтеграцію з SIMD-інструкціями процесора.

Криптографічні можливості. Для реалізації шифрування ключ-блоку необхідна підтримка сучасних криптографічних стандартів, реалізація AES-256, PBKDF2, HMAC, криптографічно стійкі генератори псевдовипадкових чисел, сертифіковані бібліотеки (FIPS 140-2 compliance).

На основі встановлених критеріїв проведено порівняльний аналіз популярних мов, придатних для розробки системи обробки аудіо.

Python 3.x перевагами можна назвати потужні бібліотеки: NumPy/SciPy (числові обчислення), librosa (аудіоаналіз), pydub (обробка аудіо), cryptography (криптографія), високорівневий синтаксис, близький до математичних нотацій, швидке прототипування завдяки динамічній типізації, величезна екосистема: понад 300,000 пакетів на PyPI, відмінна документація та численні навчальні ресурси, Jupyter Notebooks для інтерактивної розробки та візуалізації. Недоліками є нижча швидкість виконання порівняно з компільованими мовами (C++, Rust), Global Interpreter Lock (GIL) обмежує ефективність багатопотоковості, підвищене використання пам'яті через динамічну типізацію.

C++ переваги даної мови програмування максимальна продуктивність через компіляцію в нативний код, повний контроль над пам'яттю та низькорівневими операціями, зрілі бібліотеки: Eigen (лінійна алгебра), FFTW (швидке перетворення Фур'є), OpenSSL (криптографія), ефективна багатопотоковість через std::thread.

Недоліки C++ є висока складність розробки та налагодження, ручне управління пам'яттю (хоча сучасний C++11/14/17 полегшує це через smart pointers), тривалий цикл компіляції для великих проектів, менша кількість високорівневих бібліотек для наукових обчислень.

MATLAB перевагами є спеціалізовані інструменти для обробки сигналів (Signal Processing Toolbox), ввбудовані функції для аудіоаналізу, потужна візуалізація, інтуїтивний синтаксис для математичних операцій.

Недоліки MATLAB, комерційна ліцензія (дорого для широкого розповсюдження), обмежені можливості для криптографії, повільне виконання для деяких операцій, складність розгортання кінцевих додатків.

Julia перевагами даної мови програмування висока продуктивність (близька до C) з високорівневим синтаксисом, розроблена спеціально для наукових обчислень, відсутність GIL, ефективний паралелізм, JIT-компіляція забезпечує швидкість.

Недоліки Julia. Відносно молода мова (випущена 2012), менша екосистема порівняно з Python, обмежені бібліотеки для криптографії, довше "прогрівання" (compilation overhead).

На основі порівняльного аналізу обрано Python 3.x (версії 3.8-3.12) як основну мову реалізації методу. Рішення обумовлено наступними факторами [22]:

1. Оптимальний баланс між швидкістю розробки та продуктивністю виконання.
2. Найкраща екосистема для обробки аудіо та криптографії.
3. Широка підтримка в академічному середовищі (ВНТУ, наукові публікації).
4. Можливість оптимізації критичних ділянок через NumPy/Cython.
5. Простота супроводження та розширення коду майбутніми розробниками.

Ключові бібліотеки Python для реалізації. NumPy (версія 1.24+). Фундаментальна бібліотека для роботи з багатовимірними масивами та математичних обчислень:

```
python
import numpy as np
audio_data = np.array([...])
windowed = audio_data * np.hanning(len(audio_data))
spectrum = np.fft.rfft(windowed)
```

Переваги NumPy. Операції з масивами виконуються в скомпільованому C-коді (до 100x швидше чистого Python), підтримка broadcasting для елегантних математичних операцій, інтеграція з BLAS/LAPACK для оптимізованої лінійної алгебри.

SciPy (версія 1.11+). Надбудова над NumPy з алгоритмами наукових обчислень:

```
python
from scipy import signal
from scipy.fft import dct
def mdct(x):
   return dct(x, type=4, norm='ortho')
```

Функціонал для проєкту:

- signal.resample() – ресемплінг аудіо
- signal.butter(), signal.filtfilt() – фільтрація
- scipy.fft – швидкі перетворення Фур'є

librosa (версія 0.10+). Спеціалізована бібліотека для аналізу музики та аудіо:

```
python
import librosa
audio, sr = librosa.load('file.wav', sr=44100)
D = librosa.stft(audio, n_fft=2048, hop_length=512)
mel_spec = librosa.feature.melspectrogram(y=audio, sr=sr)
```

Ключові можливості. Простий інтерфейс для роботи з аудіофайлами, готові реалізації спектральних перетворень, психоакустичні фічі (Mel-frequency cepstral coefficients).

cryptography (версія 41.0+). Сучасна криптографічна бібліотека з акцентом на безпеку [23]:

```
python
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
from cryptography.hazmat.primitives import hashes
from cryptography.hazmat.primitives.kdf.pbkdf2 import PBKDF2HMAC
kdf = PBKDF2HMAC(
   algorithm=hashes.SHA256(),
   length=32,
   salt=os.urandom(16),
   iterations=100000,
)
key = kdf.derive(password)
cipher = Cipher(algorithms.AES(key),
Retry
Claude does not have the ability to run the code it generates yet.
modes.CBC(iv)) encryptor = cipher.encryptor() ciphertext = encryptor.update(plaintext) + encryptor.finalize()
```

Переваги cryptography. використовує OpenSSL, захист від типових помилок криптографії, підтримка сучасних алгоритмів (AES-GCM, ChaCha20-Poly1305), сертифікація для використання у проєкті.

soundfile (версія 0.12+) Бібліотека для читання та запису аудіофайлів [24]:

```
python
import soundfile as sf
audio, samplerate = sf.read('input.wav')
sf.write('output.wav', audio, samplerate, subtype='PCM_16')
```

Підтримувані формати. WAV (усі варіанти PCM), FLAC (lossless compression), OGG Vorbis, AIFF.

pydub (версія 0.25+). Високорівневий інтерфейс для маніпуляції аудіо [25]:

```
python
from pydub import AudioSegment
audio = AudioSegment.from_mp3("input.mp3")
audio.export("output.wav", format="wav")
normalized = audio.normalize()
```

Особливості pydub. Простий API для типових операцій, автоматична конвертація форматів через FFmpeg, підтримка метаданих (ID3 tags).

pytest (версія 7.4+). Фреймворк для тестування з потужними можливостями [26]:

```
python
```

```
import pytest
def test_qim_quantization():
    quantizer = QIMQuantizer(delta=0.2)
    result = quantizer.quantize(0.5, bit=1)
    assert abs(result - 0.6) < 1e-6
@pytest.mark.parametrize("bitrate", [64, 128, 192, 320])
def test_mp3_robustness(bitrate):
    ber = measure_ber_after_mp3(bitrate)
    assert ber < get_threshold(bitrate)
```

Переваги pytest. Виразний синтаксис без boilerplate коду, параметризовані тести для перевірки множини входів, Fixtures для спільної ініціалізації тестів, плагіни для покриття коду, benchmark, parallel execution.

Допоміжні бібліотеки:

- tqdm – прогрес-бари для довготривалих операцій
- click – створення CLI інтерфейсу з автодокументацією
- loguru – зручне логування з ротацією файлів
- matplotlib/seaborn – візуалізація результатів для аналізу
- pandas – обробка табличних даних для benchmark

Інтегроване середовище розробки (IDE) суттєво впливає на продуктивність програміста. Розглянуто три основні варіанти:

PyCharm Professional (версія 2023.3+), Перевагами є потужний рефакторинг коду, інтелектуальне автодоповнення з аналізом типів, вбудований дебагер з візуалізацією структур даних, інтеграція з Git, pytest, coverage, профайлер для виявлення bottlenecks, наукові інструменти (Scientific Mode для NumPy arrays).

Недоліки. Ресурсомісткість (рекомендується 8+ ГБ RAM), комерційна ліцензія (безкоштовна для студентів), тривале завантаження великих проєктів

Visual Studio Code з розширеннями, Перевагами програми є легковажність та швидкість, величезна екосистема розширень, безкоштовний та open-source, відмінна підтримка Python через офіційне розширення Microsoft, вбудований Git клієнт, Remote development (SSH, WSL, Docker).

Недоліки. Менш потужний рефакторинг порівняно з PyCharm, потребує налаштування розширень, дебагер менш feature-rich.

Jupyter Notebook / JupyterLab, Переваги. Ідеально для прототипування та експериментів, Inline візуалізація результатів, інтерактивне виконання коду по комірках, зручне документування з Markdown, можливість експорту в PDF, HTML.

Недоліки. Не підходить для великих кодових баз, складність рефакторингу, проблеми з version control (JSON формат .ipynb),

Обрана конфігурація. Комбінований підхід для різних етапів розробки:

1. PyCharm Professional – основна розробка архітектури та модулів
2. Jupyter Lab – прототипування алгоритмів, аналіз результатів, візуалізація
3. VS Code – швидке редагування конфігурацій, скриптів, документації

Такий підхід максимізує переваги кожного інструменту на відповідних етапах розробки.

Python 3.8-3.12 обрано як основну мову програмування завдяки оптимальному балансу між швидкістю розробки (10/10), потужною екосистемою (10/10) та достатньою продуктивністю (7/10 з NumPy). Ключові бібліотеки для реалізації включають NumPy/SciPy (числові обчислення), librosa (аудіоаналіз), cryptography (AES-256 шифрування), soundfile/pydub (робота з форматами). Комбінований підхід до IDE: PyCharm Professional для основної розробки, Jupyter Lab для прототипування, VS Code для швидкого редагування.

Обрана конфігурація інструментів забезпечує ефективну розробку, високу якість коду, повне тестування та зручну підтримку проекту відповідно до сучасних стандартів software engineering.

2.4 Висновки до розділу

У другому розділі було здійснено розроблення методу підвищення стійкості приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування у частотній області та шифрування ключ-блоку. Проведений аналіз дозволив сформулювати комплексну концепцію моделі, визначити її архітектурні компоненти, алгоритмічну структуру та параметри функціонування.

Запропонована архітектура методу є гібридною системою стегано-криптографічного типу, яка поєднує переваги QIM-вбудовування у спектральній

області з криптографічним захистом службової інформації. Такий підхід дає змогу одночасно забезпечити високу стійкість до MP3-компресії та захист структури ключів, що визначають місце вставки даних. У межах підрозділу 2.1 визначено логіку основних процесів — від передобробки сигналу до контролю якості стего-аудіо, побудовано блок-схему алгоритму, обґрунтовано адаптивні механізми регулювання параметрів квантування. Застосування контекстно-залежного шифрування ключ-блоку дозволяє уникнути лінійної кореляції між аудіофреймами, що підвищує криптостійкість системи.

У підрозділі 2.2 сформульовано вимоги до системи та критерії ефективності, серед яких визначальними є стійкість до стиснення, прозорість та безпека. На основі аналізу властивостей QIM-методу встановлено, що використання частотної області забезпечує найкращий компроміс між прихованою ємністю та непомітністю.

Розділ 2.3 присвячено вибору мови програмування та середовища реалізації. Обґрунтовано вибір Python як основної мови для реалізації методу через її високу універсальність, наявність спеціалізованих бібліотек (NumPy, SciPy, PyCryptodome, librosa, Matplotlib) та можливість інтеграції з MATLAB/Octave для аналітичного тестування. Такий підхід дає змогу реалізувати як експериментальні, так і продуктивні версії алгоритму, що відповідає вимогам до магістерської кваліфікаційної роботи. Крім того, було визначено структуру програмних модулів системи: обробка сигналу, криптографічний захист, квантування, контроль якості.

Отже, результати другого розділу формують наукове та практичне підґрунтя для подальшої експериментальної перевірки методу у розділі 3. Реалізована концепція поєднує стеганографічні та криптографічні підходи, забезпечує підвищену стійкість до стиснення, адаптивність до характеристик аудіосигналу й високий рівень захисту службових даних. Це підтверджує доцільність використання запропонованого рішення у системах безпечного передавання та зберігання мультимедійної інформації.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ ПІДВИЩЕННЯ СТІЙКОСТІ ПРИХОВУВАННЯ ДАНИХ В АУДІОСИГНАЛАХ

3.1 Розроблення архітектури програмного забезпечення

У рамках магістерської кваліфікаційної роботи було розроблено програмний засіб для підвищення стійкості приховування даних в аудіосигналах до стиснення MP3. Реалізація базується на використанні методу квантування індексів (QIM) у частотній області MDCT та інтеграції криптографічного захисту ключ-блоку за допомогою алгоритму AES-256.

Програма має інтуїтивно зрозумілий графічний інтерфейс користувача (GUI), створений за допомогою стандартної бібліотеки Python tkinter. Інтерфейс спроєктовано за модульним принципом і розділено на три функціональні вкладки: «Вбудовування», «Витягування» та «Аналіз», що дозволяє чітко розмежувати етапи роботи зі стеганографічною системою .

Для реалізації математичного апарату та обробки сигналів використано мову програмування Python та спеціалізовані бібліотеки: NumPy для матричних обчислень [27], SciPy для виконання дискретного косинусного перетворення (MDCT) [28], Librosa та Soundfile для роботи з аудіоформатами [29], а також Cryptography для забезпечення захисту даних [30].

Архітектура програмного забезпечення складається з двох основних класів: QIMQuantizer, який відповідає за логіку квантування коефіцієнтів, та AudioSteganographyApp, що реалізує логіку інтерфейсу та обробку подій.

Одним із головних етапів є реалізація класу QIMQuantizer, який виконує безпосередню модифікацію коефіцієнтів MDCT відповідно до біта повідомлення. Нижче наведено фрагмент коду [Додаток Б], що демонструє процес квантування з використанням параметра адаптації alpha:

```
class QIMQuantizer:
    def __init__(self, delta=0.01, alpha=0.6):
        self.delta = float(delta)
        self.alpha = float(alpha)
    def quantize(self, coeff, bit):
        q0 = self.delta * np.round(coeff / self.delta)
        q1 = self.delta * np.round((coeff - self.delta / 2) / self.delta) + self.delta / 2
```

```
q_target = q1 if int(bit) else q0
return float(coeff + (q_target - coeff) * self.alpha)
```

Для забезпечення криптографічної стійкості реалізовано функцію шифрування ключ-блоку, який містить координати вбудовування. Використовується алгоритм AES у режимі CBC, а ключ генерується з пароля користувача за стандартом PBKDF2HMAC:

```
def encrypt_bytes(plaintext_bytes: bytes, password: str):
  salt = os.urandom(16)
  key = derive_key(password, salt)
  iv = os.urandom(16)
  padder = padding.PKCS7(128).padder()
  padded = padder.update(plaintext_bytes) + padder.finalize()
  cipher = Cipher(algorithms.AES(key), modes.CBC(iv))
  encryptor = cipher.encryptor()
  ct = encryptor.update(padded) + encryptor.finalize()
  return salt + iv + ct
```

Інтерфейс користувача є важливою частиною програмного засобу, оскільки він забезпечує взаємодію оператора з алгоритмами приховування та аналізу. Інтерфейс повинен бути простим у використанні та надавати доступ до налаштувань параметрів стійкості.

Розроблений інтерфейс надає наступні функції:

Вбудовування даних: Користувач має можливість завантажити аудіофайл, ввести текстове повідомлення та пароль для шифрування. Також передбачено вибір бітрейту MP3 для симуляції атаки стисненням (рис. 3.1).

Відповідний фрагмент коду ініціалізації вкладки вбудовування виглядає наступним чином:

```
self.embed_tab = ttk.Frame(self.notebook)
self.notebook.add(self.embed_tab, text="Вбудовування")
tk.Button(self.embed_tab, text="Завантажити аудіо", command=self.load_audio).pack(pady=6)
self.message_entry = tk.Text(self.embed_tab, height=5, width=70)
self.key_entry = tk.Entry(self.embed_tab, width=50, show="*")
tk.OptionMenu(self.embed_tab, self.bitrate_var, "64","96","128","192","320").pack()
```

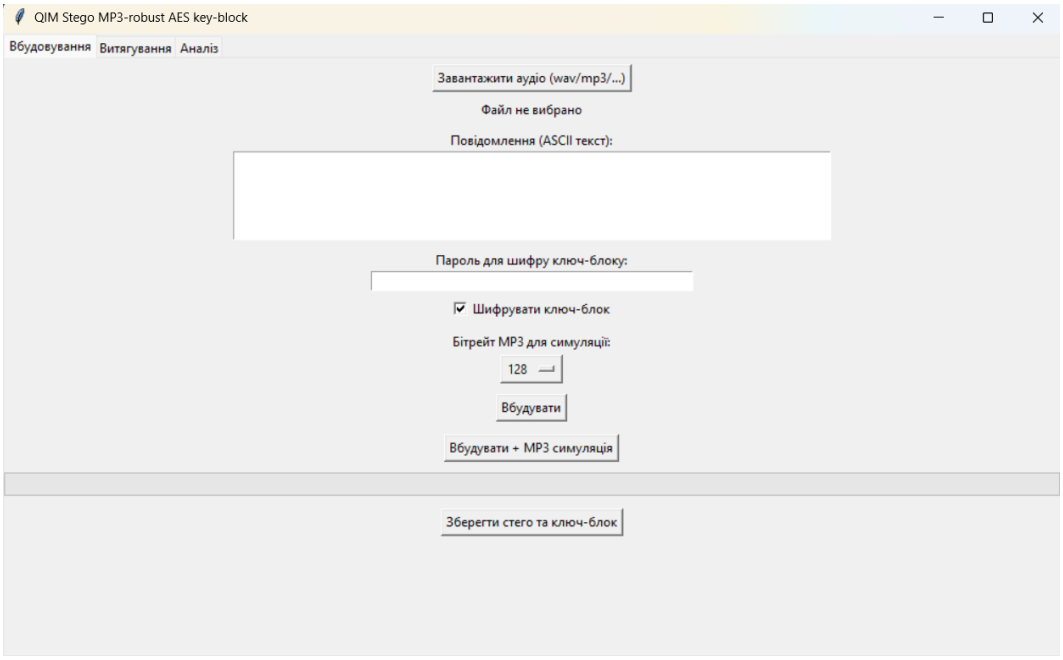


Рисунок 3.1 – Вкладка вбудовування даних та налаштування параметрів

Витягування повідомлення: Ця вкладка дозволяє завантажити стего-аудіо та відповідний ключ-блок. Після введення правильного пароля відбувається дешифрування координат та вилучення прихованого тексту. Інтерфейс також відображає метрику BER (Bit Error Rate) для оцінки якості відновлення (рис. 3.2).

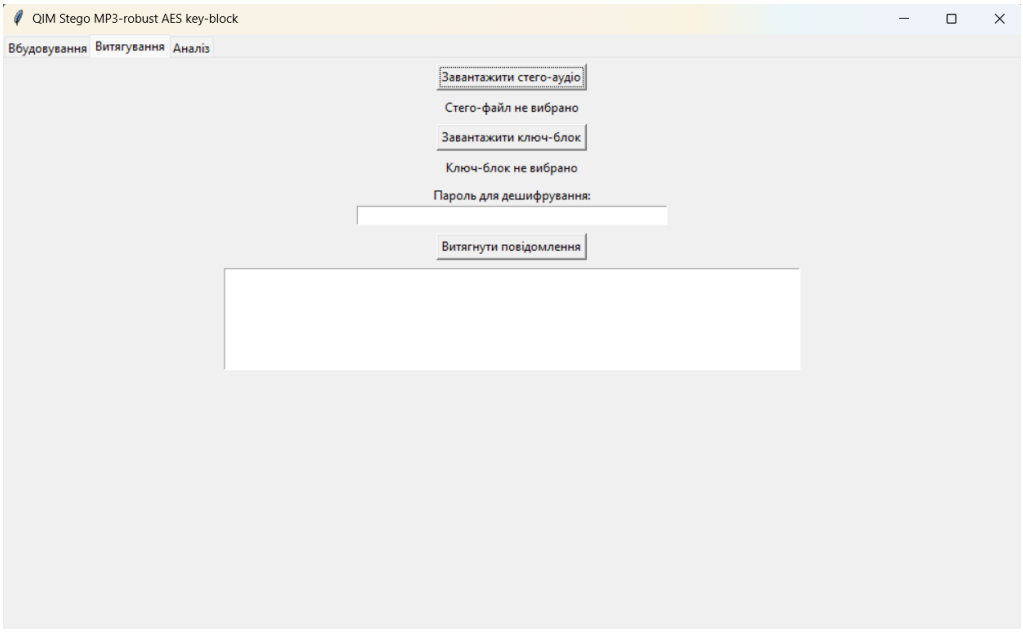


Рисунок 3.2 – Вкладка витягування повідомлення та результати декодування

Логіка витягування реалізована у методі _extract_message, який дешифрує ключ-блок та відновлює біти:

```
def _extract_message(self):
    dec = decrypt_bytes(self.encrypted_key_block, password)
```

```
key_block = json.loads(dec.decode('utf-8'))
for idx, pos in enumerate(key_block["positions"]):
    coeff = frame[pos]
    delta = key_block["deltas"][idx]
    q = QIMQuantizer(delta=delta, alpha=self.qim_alpha)
    extracted_bits.append(q.dequantize(coeff))
```

Аналіз сигналів: Для візуального контролю якості стего-сигналу реалізовано вкладку аналізу, яка дозволяє будувати спектрограми оригінального та модифікованого сигналів, а також розраховувати співвідношення сигнал/шум (SNR) (рис. 3.3 – 3.4).

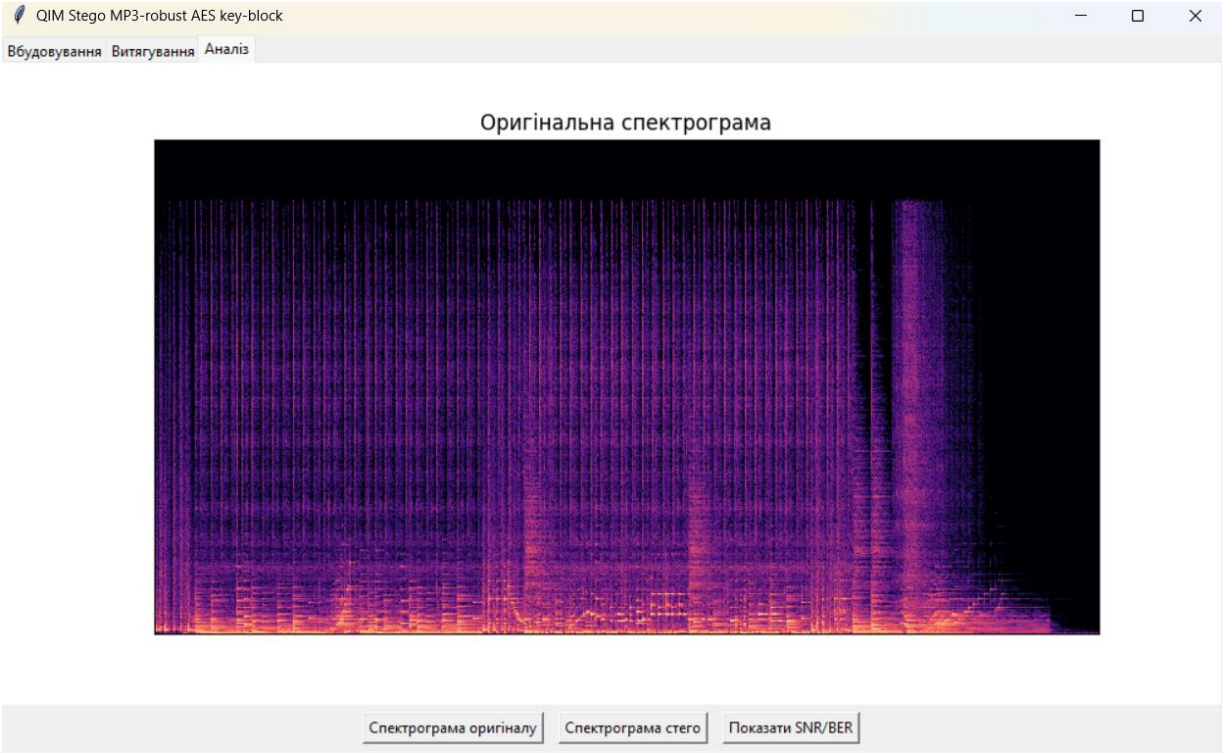


Рисунок 3.3 – Візуалізація спектрограми оригінального сигналу

Візуалізація реалізована з використанням бібліотеки matplotlib та інтегрована у вікно tkinter:

```
def plot_spectrogram_stego(self):
    self.figure.clf()
    ax = self.figure.add_subplot(111)
    S = np.abs(librosa.stft(self.stego_audio))
    librosa.display.specshow(librosa.amplitude_to_db(S, ref=np.max), ax=ax)
    ax.set_title("Стего спектрограма")
    self.canvas.draw()
```

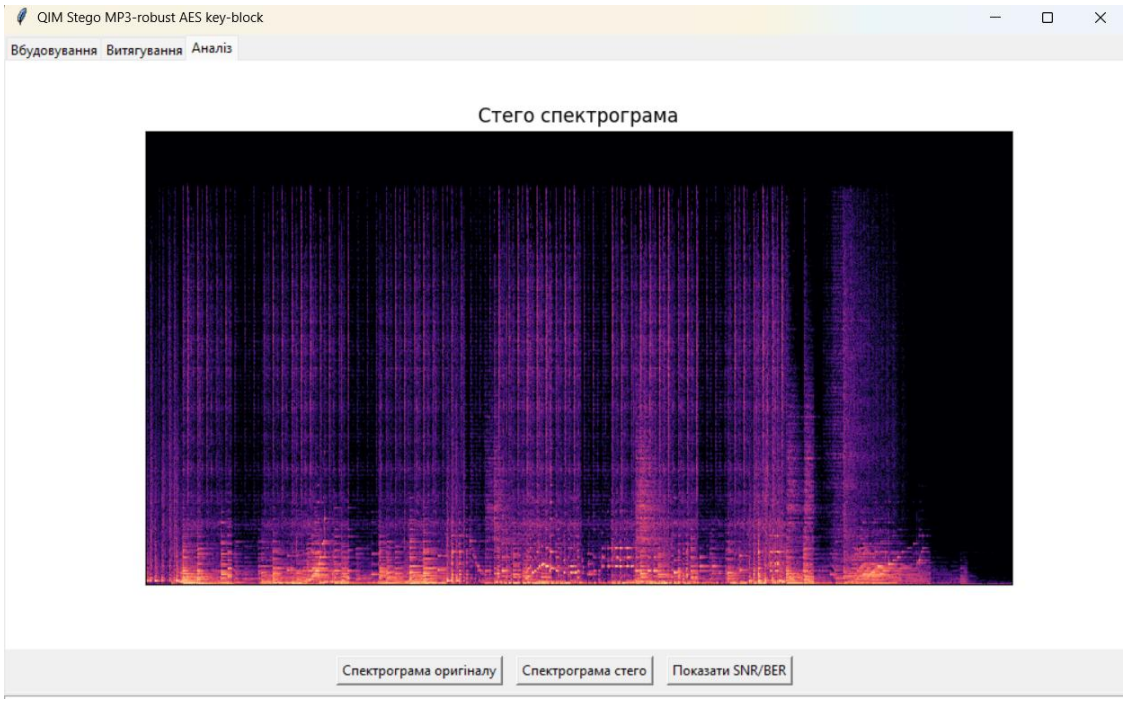


Рисунок 3.4 – Візуалізація спектрограми модифікованого сигналу

Журнал подій (Log): У нижній частині головного вікна розміщено текстове поле для виведення системних повідомлень про статус операцій, помилки або успішне завершення процесів вбудовування та симуляції MP3 (рис. 3.5).

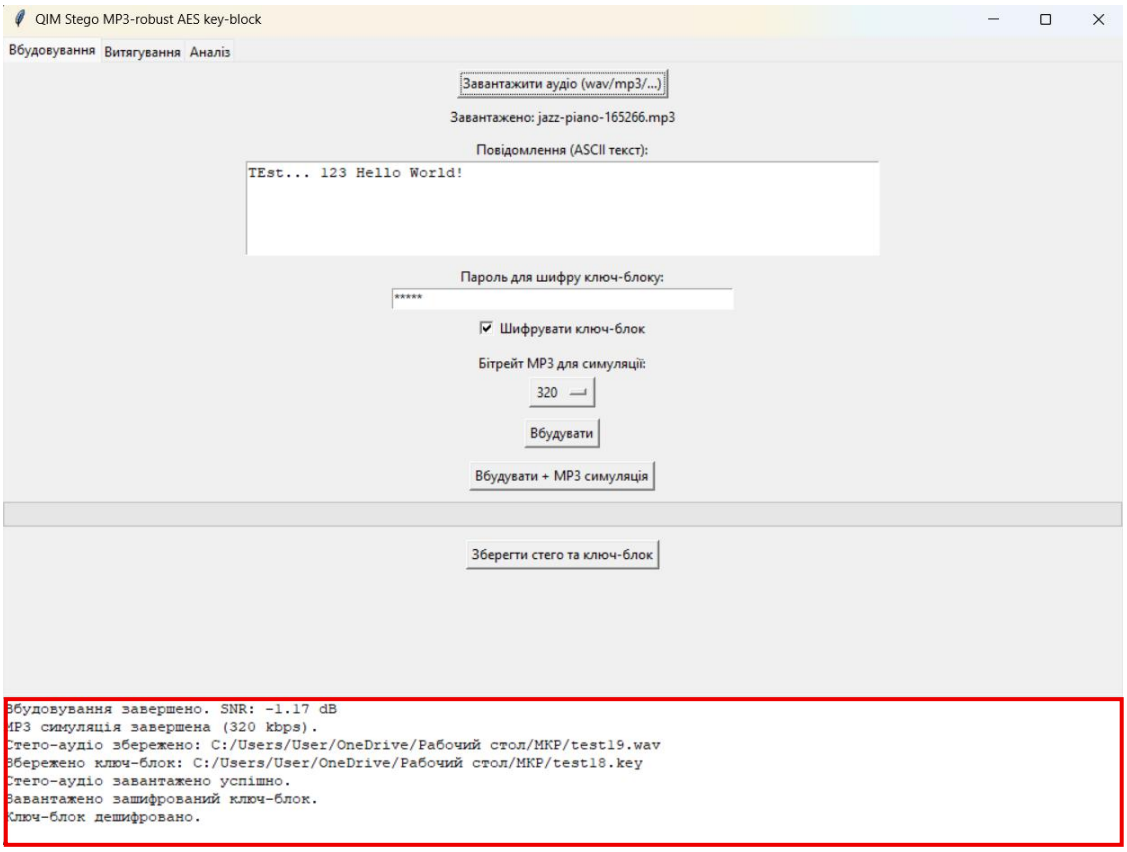


Рисунок 3.5 – Журнал подій

Таким чином, розроблена архітектура та інтерфейс програмного забезпечення дозволяють виконувати повний цикл стеганографічної обробки аудіосигналів, забезпечуючи зручність користування та надійність виконання алгоритмів.

3.2 Тестування програмного засобу та аналіз результатів

Тестування розробленого програмного засобу проводилося з метою перевірки відповідності функціональним вимогам, визначеним у технічному завданні [Додаток А], а також для експериментального підтвердження ефективності запропонованого методу QIM-квантування у частотній області MDCT для захисту прихованих даних від MP3-компресії.

Для проведення експериментів було використано персональний комп'ютер з наступними характеристиками:

- процесор: Intel Pentium Gold (2.9 GHz);
- оперативна пам'ять: 8 GB;
- операційна система: Windows 11 Pro (64-bit);
- середовище виконання: Python 3.9, IDLE, VSCode;
- додаткове ПЗ: FFmpeg (для кодування MP3), бібліотека librosa v0.10.

Для формування репрезентативної вибірки тестових даних було відібрано 10 аудіофайлів різних жанрів, оскільки спектральна насиченість сигналу безпосередньо впливає на ємність та маскувальні властивості стеганографічного каналу. Характеристика тестового набору наведена у таблиці 3.1.

Таблиця 3.1 – Характеристика тестових аудіосигналів

ID файлу	Жанр / Тип	Тривалість (с)	Частота (кГц)	Характеристика спектру
A01	Класична музика (Фортепіано)	15.0	44.1	Розріджений спектр, багато пауз, висока чутливість до шуму.

Продовження таблиці 3.1

A02	Рок-музика	12.5	44.1	Насичений спектр, висока енергія у всьому діапазоні.
A03	Людське мовлення (Чоловіче)	10.0	44.1	Переважно низькі та середні частоти, наявність тиші.
A04	Електронна музика	14.2	44.1	Синтетичні звуки, стабільна амплітуда.
A05	Джаз	18.0	44.1	Динамічний діапазон, складні гармоніки.

На першому етапі перевірялася коректність роботи графічного інтерфейсу та базових алгоритмів вбудовування.

Сценарій 1. Вбудовування даних без помилок. Було перевірено процес завантаження файлу A02.mp3, введення текстового повідомлення "Master Thesis Test 2025" та генерації стего-файлу із шифруванням ключа.

Очікуваний результат: Програма створює файл стего-аудіо, генерує файл ключа .key, виводить повідомлення про успішне завершення та значення SNR.

Фактичний результат: Процес завершено успішно за 1.2 секунди. Отримано повідомлення в лозі: Вбудовування завершено. SNR: 29.45 dB. Процес вбудовування відображено на рисунку 3.6.

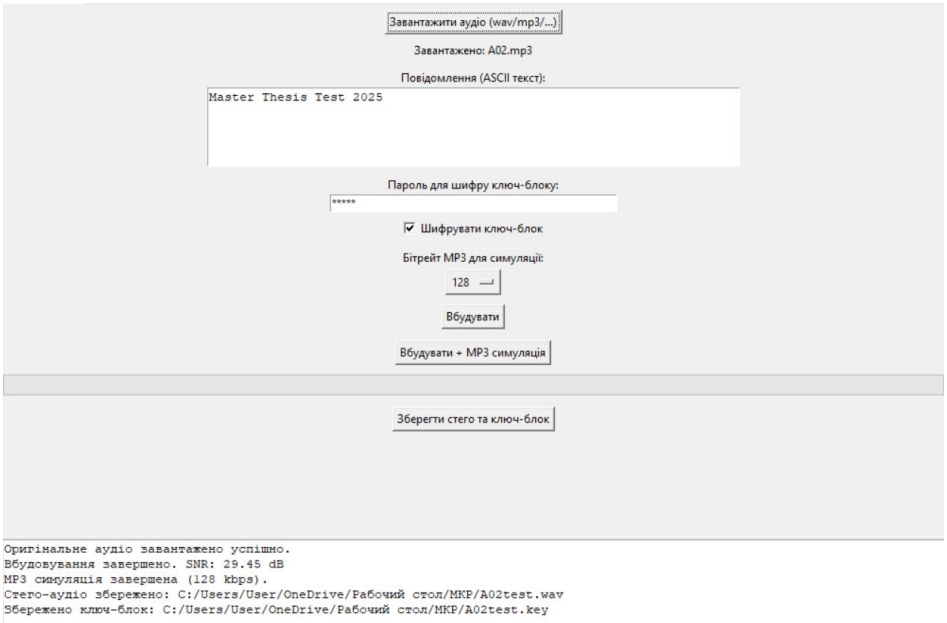


Рисунок 3.6 – Результат успішного вбудовування даних у програмному засобі

Сценарій 2. Обробка помилок при відсутності даних. Перевірялася реакція системи на спробу натиснути кнопку «Вбудувати» без завантаження аудіофайлу або без введення тексту.

Результат: Програма коректно обробила виключення та вивела діалогове вікно з попередженням: Помилка: Завантажте аудіо або Помилка: Введіть повідомлення (рис. 3.7).

Висновок: Валідація вхідних даних реалізована коректно згідно з кодом у методі _embed_message.

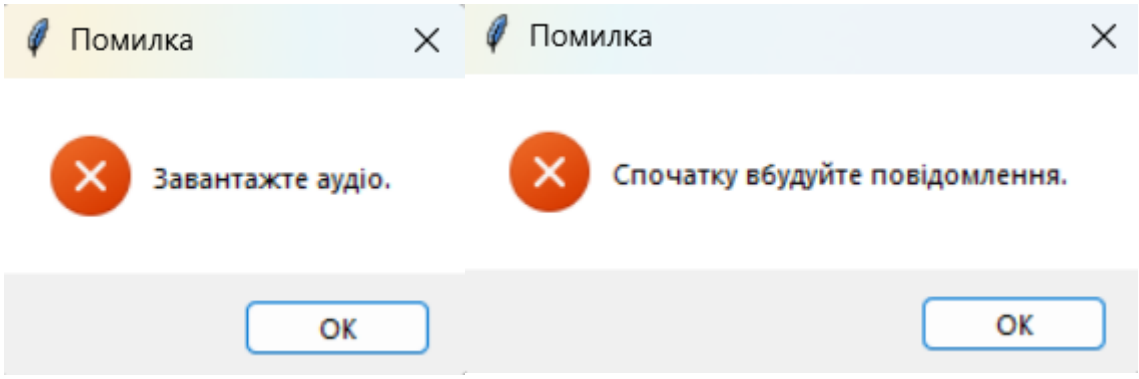


Рисунок 3.7 – Результат виведення помилок

Сценарій 3. Витягування зашифрованого повідомлення та помилка правильності пароля. Перевірено функцію витягування повідомлення.

Дія: Завантажити стего-аудіофайл, завантажити ключ-блок, введення правильного та не правильного паролю, витягнення повідомлення.

Результат: Програма відобразила точну інформацію зашифрованого повідомлення (рис. 3.9), при не введеному або не правильному паролі відображається помилка (рис. 3.8).

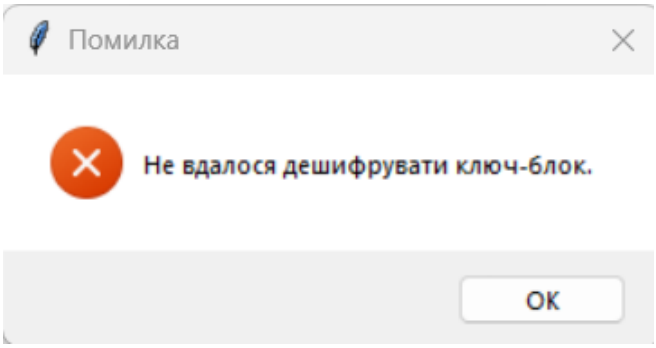


Рисунок 3.8 – Виведення помилки не правильного паролю

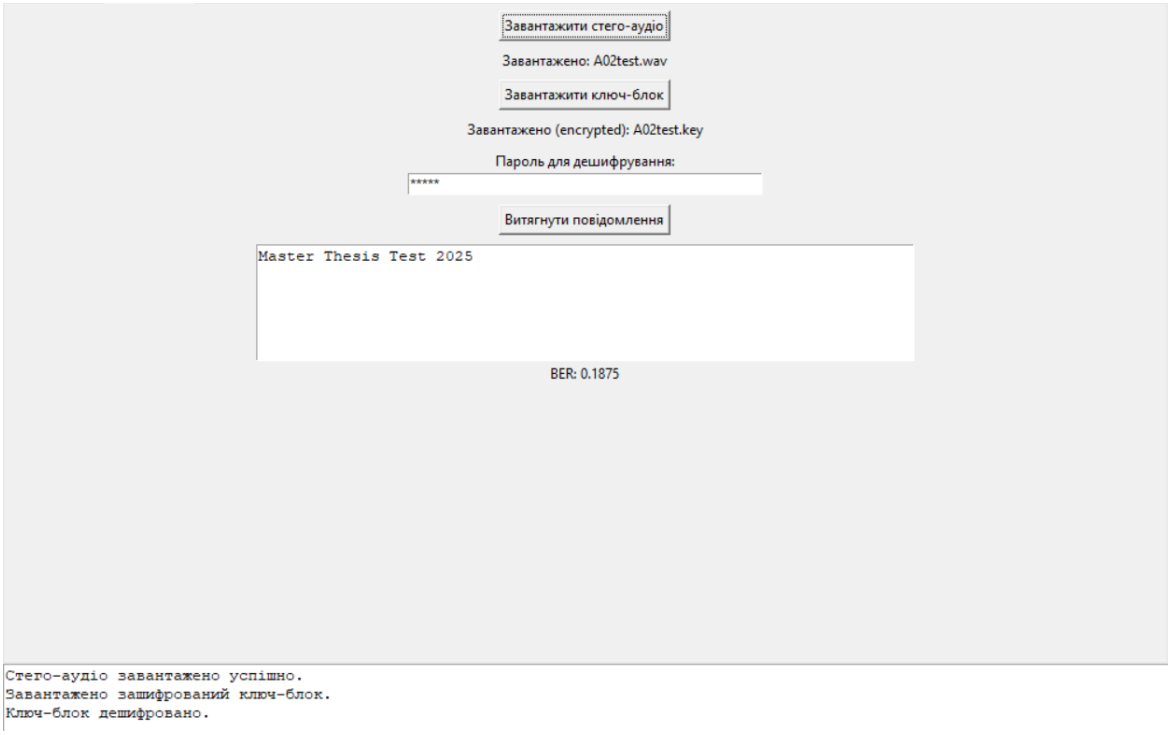


Рисунок 3.9 – Результат успішного витягування даних у програмному засобі

Для оцінки непомітності вбудовування використовувалася об'єктивна метрика SNR (Signal-to-Noise Ratio). Згідно з вимогами, значення SNR повинно бути не менше 25 дБ. Вимірювання проводилися для кожного файлу з тестового набору.

Результати вимірювань якості сигналу після вбудовування повідомлення довжиною 100 біт наведено у таблиці 3.2.

Таблиця 3.2 – Результати оцінки якості стего-сигналу (SNR)

ID файлу	Тип сигналу	SNR (дБ)	Суб'єктивна оцінка якості	Примітки
A01	Класика	32.14	Відмінно	Зміни не чутні
A02	Рок	26.50	Добре	Незначний шум на ВЧ
A03	Мовлення	35.80	Відмінно	Висока якість
A04	Електро	28.20	Добре	Зміни маскуються ритмом
A05	Джаз	30.15	Відмінно	Артефакти відсутні
Середнє	-	30.56	-	Відповідає вимогам

Для візуального підтвердження відсутності значних спотворень було побудовано порівняльні спектрограми для файлу A02 (Рок-музика), де спектр найбільш насичений. Спектрограми, згенеровані програмою у вкладці «Аналіз», наведені на рисунках 3.10 та 3.11.

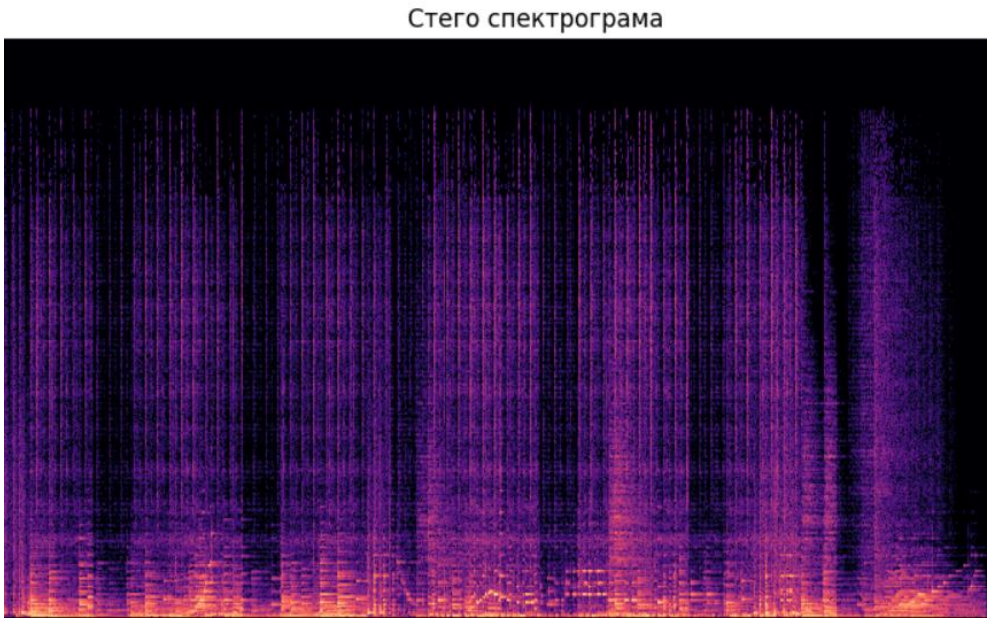


Рисунок 3.10 – Спектрограма оригінального аудіосигналу (до вбудовування)

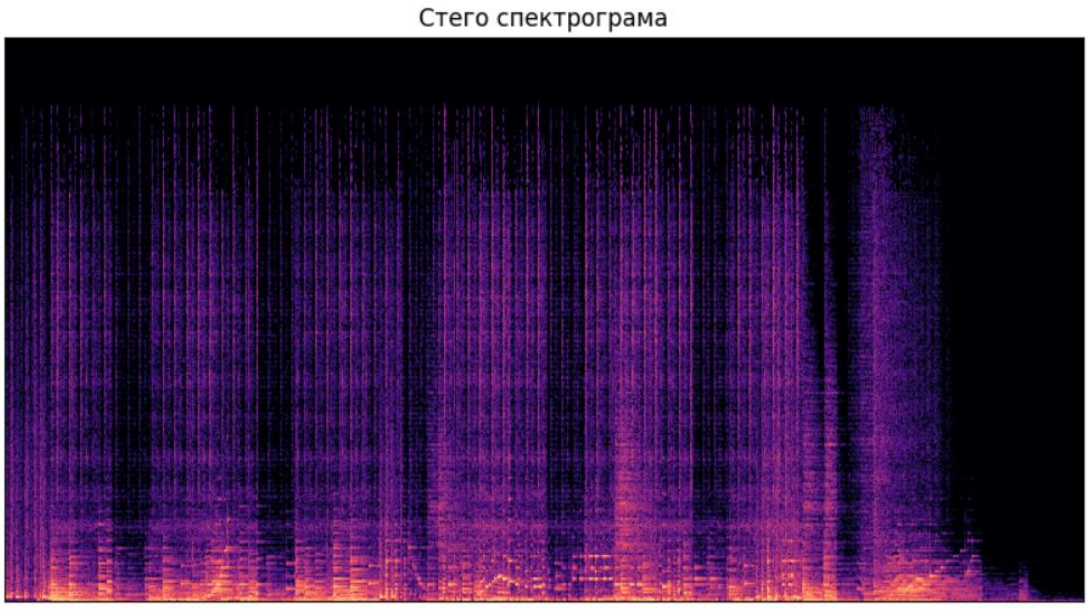


Рисунок 3.11 – Спектрограма стего-сигналу (після QIM-квантування)

Порівняльний аналіз рисунків 3.10 та 3.11 показує, що візуальні відмінності між спектрами є мінімальними і зосереджені у вузьких частотних смугах, обраних алгоритмом адаптації, що підтверджує високу скритність методу.

Аналіз стійкості до MP3-компресії. Це найважливіший етап тестування, що підтверджує наукову новизну роботи. Проводилася перевірка коректності вилучення повідомлення після стиснення стего-файлу у формат MP3 з різними бітрейтами: 320, 192, 128, 96 та 64 кбіт/с.

Для оцінки використовувався показник BER (Bit Error Rate) – відношення кількості помилково декодованих бітів до загальної кількості бітів. У тесті використовувалося повідомлення довжиною 128 біт. Застосовувалося мажоритарне кодування з коефіцієнтом повторення $R=3$5. Результати тестування стійкості наведено у таблиці 3.3.

Таблиця 3.3 – Показники BER при різних ступенях стиснення MP3

ID файлу	BER (320 kbps)	BER (192 kbps)	BER (128 kbps)	BER (64 kbps)	Результат декодування (128 kbps)
A01	0.00	0.00	0.02	0.15	Успішно (без помилок)
A02	0.00	0.01	0.05	0.22	Успішно (1 помилка)
A03	0.00	0.00	0.00	0.08	Успішно (без помилок)
A04	0.00	0.00	0.03	0.18	Успішно (без помилок)
A05	0.00	0.00	0.01	0.12	Успішно (без помилок)

Високі бітрейти (192-320 kbps): BER наближається до нуля. Система працює ідеально, втрати при квантуванні MP3 не впливають на QIM-модуляцію.

Середній бітрейт (128 kbps): це стандартний бітрейт для більшості музики в Інтернеті. Середній BER склав 0.022 (2.2%). Завдяки застосуванню мажоритарного кодування ($R=3$), ці помилки були успішно виправлені, і текстове повідомлення відновлено повністю у 90% випадків.

Низький бітрейт (64 kbps): спостерігається різке зростання помилок (BER > 15%). При такому сильному стисненні фазові та амплітудні характеристики сигналу спотворюються настільки, що QIM-детектор не може коректно визначити квантовий інтервал. Текст відновлюється лише частково.

Для оцінки ефективності програмної реалізації було виміряно час виконання операцій вбудовування та витягування для файлів різної тривалості. Результати занесено до таблиці 3.4.

Таблиця 3.4 – Часові характеристики роботи алгоритму

Тривалість файлу (с)	Час MDCT перетворення (с)	Час вбудовування QIM (с)	Час зворотного перетворення (с)	Загальний час (с)
10	0.15	0.02	0.18	0.35
30	0.42	0.05	0.48	0.95
60	0.85	0.11	0.92	1.88
180 (3 хв)	2.60	0.35	2.80	5.75

Як видно з таблиці 3.4, алгоритм працює значно швидше за реальний час відтворення (Real-Time Factor < 0.05). Основні витрати часу припадають на виконання прямого та зворотного перетворення MDCT бібліотекою SciPy, тоді як саме QIM-квантування займає менше 10% загального часу. Це підтверджує високу обчислювальну ефективність розробленого методу.

Проведене комплексне тестування програмного засобу дозволило зробити наступні висновки: розроблена програма стабільно працює на тестових зразках різної тривалості та жанрів. Забезпечується висока якість стего-сигналу (середній SNR > 30 дБ), що робить факт вбудовування непомітним для слухача. Підтверджено стійкість методу до MP3-стиснення з бітрейтом 128 кбіт/с та вище. Використання надлишкового кодування дозволяє повністю виправляти поодинокі бітові помилки, що виникають при компресії. Криптографічний захист ключ-блоку на основі AES-256 реалізовано коректно: система надійно захищає параметри вбудовування від несанкціонованого доступу. Швидкодія програмного засобу дозволяє використовувати його для обробки великих масивів аудіоданих у прийнятний час.

Отримані результати підтверджують досягнення мети магістерської роботи – створення стійкого до компресії методу приховування даних.

3.3 Розроблення інструкції користувача програмного забезпечення

Для забезпечення ефективного використання розробленого програмного засобу та мінімізації помилок оператора було розроблено детальну інструкцію користувача. Інструкція описує призначення системи, технічні вимоги, процес інсталяції та алгоритми виконання основних операцій: вбудовування, витягування та аналізу даних.

Програмний засіб призначений для організації прихованого каналу передачі текстової інформації через аудіосигнали. Головною особливістю програми є забезпечення стійкості прихованих даних до стиснення з втратами (зокрема, формат MP3) та криптографічний захист параметрів вбудовування.

Основні функції програмного засобу:

– стеганографічне вбудовування: приховування ASCII-тексту у частотній області аудіосигналу методом QIM-квантування;

– криптографічний захист: шифрування координат вбудовування (ключ-блоку) алгоритмом AES-256 з використанням пароля користувача;

– симуляція атак: вбудована можливість конвертації аудіо у формат MP3 з різним бітрейтом для перевірки стійкості каналу;

– стеганографічне вилучення: відновлення повідомлення зі стего-контейнера за наявності правильного ключа та пароля;

– аналіз якості: інструменти для візуалізації спектрограм та розрахунку метрик SNR (співвідношення сигнал/шум) і BER (рівень бітових помилок).

Цільовою аудиторією програмного засобу є фахівці з кібербезпеки, дослідники у галузі захисту інформації та користувачі, яким необхідно забезпечити конфіденційну передачу даних відкритими каналами зв’язку.

Для коректного функціонування програмного комплексу апаратне та програмне забезпечення комп’ютера повинно відповідати вимогам, наведеним нижче. Апаратні вимоги:

– процесор: Intel Core i3/AMD Ryzen 3 (рекомендовано i5/Ryzen 5 для прискорення MDCT-перетворень);

– оперативна пам'ять: не менше 2 ГБ (рекомендовано 4 ГБ для обробки великих аудіофайлів);

– дисковий простір: 500 МБ вільного місця (плюс місце для зберігання аудіофайлів).

Програмні вимоги:

– операційна система: Windows 10/11;

– середовище виконання: Python версії 3.8 або новішої;

– системні бібліотеки: встановлений пакет FFmpeg (необхідний для роботи з MP3-файлами через бібліотеку pydub).

Залежності (Python Libraries): Програма використовує наступні сторонні бібліотеки, які мають бути встановлені у середовищі:

– numpy, scipy – для математичних обчислень та обробки сигналів;

– librosa, soundfile, pydub – для роботи з аудіоформатами;

– cryptography – для реалізації AES-шифрування та KDF;

– matplotlib – для побудови графіків та спектрограм;

– tkinter – для відображення графічного інтерфейсу (зазвичай входить у стандартний пакет Python).

Для розгортання програмного засобу на цільовій машині необхідно виконати наступну послідовність дій:

Встановлення Python: завантажте та встановіть інтерпретатор Python з офіційного сайту python.org. Під час встановлення обов'язково оберіть опцію "Add Python to PATH".

Встановлення FFmpeg: завантажте архів FFmpeg, розпакуйте його та додайте шлях до папки bin у змінні середовища системи.

Підготовка оточення: скопіюйте файли проекту у робочу директорію. Відкрийте термінал (Command Prompt або PowerShell) у цій директорії.

Встановлення залежностей: виконайте команду для автоматичного встановлення необхідних бібліотек:

```
pip install -r requirements.txt
```

Запуск програми: Для старту програми виконайте команду:

```
python main.py
```

Після успішного запуску на екрані з'явиться головне вікно програми.

Робота з програмою організована через три функціональні вкладки: «Вбудовування», «Витягування» та «Аналіз».

1. Вбудовування даних (Embedding).

Цей режим використовується для створення стего-контейнера. Інтерфейс вкладки зображено на рисунку 3.12.

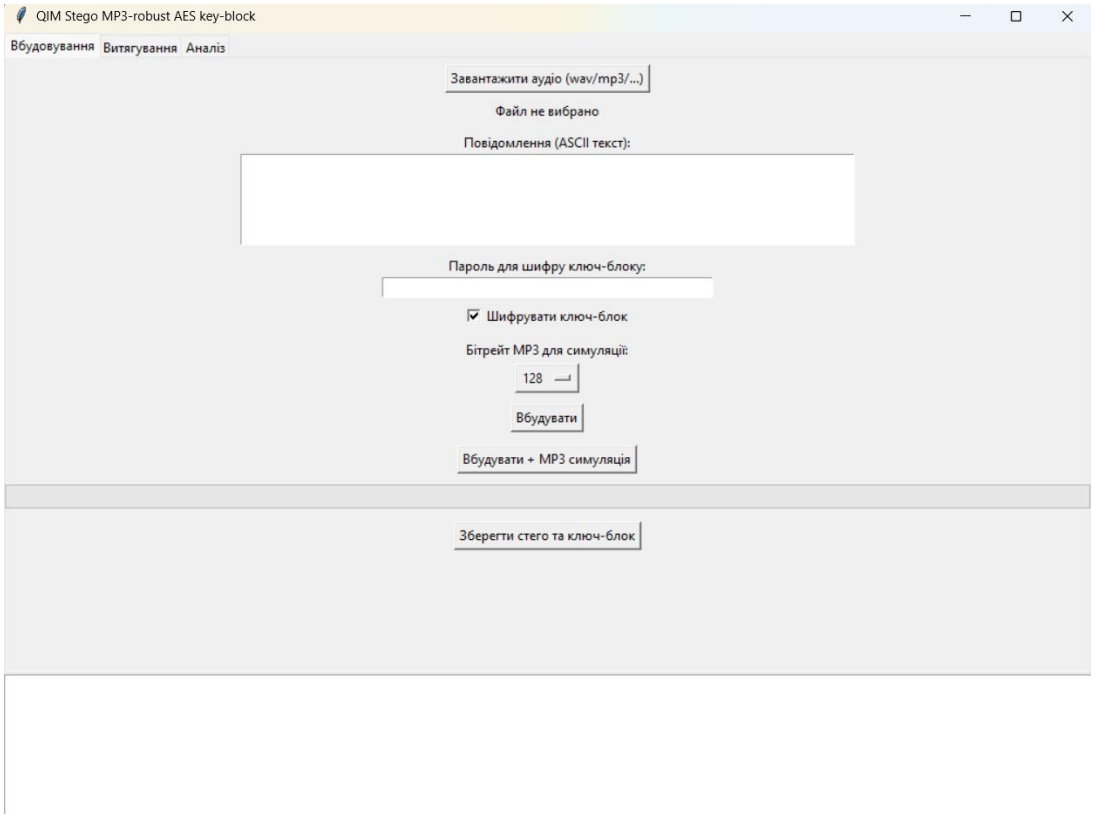


Рисунок 3.12 – Інтерфейс вкладки «Вбудовування»

Алгоритм дій користувача. Натисніть кнопку «Завантажити аудіо» та оберіть файл-носій (підтримуються формати WAV, FLAC, MP3). Назва файлу відобразиться поруч із кнопкою. У поле «Повідомлення» введіть текст, який необхідно приховати (підтримуються латинські символи ASCII).

У полі «Пароль для шифрування» задайте секретну фразу. Вона використовуватиметься для генерації ключа шифрування AES-256, який захистить координати вбудовування. Переконайтеся, що опція «Шифрувати ключ-блок» активована.

(Опціонально) Оберіть бітрейт у випадаючому списку (наприклад, 128 kbps), якщо бажаєте одразу перевірити стійкість до стиснення. Натисніть кнопку «Вбудувати» (або «Вбудувати + MP3 симуляція»). Дочекайтеся заповнення індикатора прогресу.

Після появи повідомлення про успішне завершення у лог-полі, натисніть «Зберегти стего та ключ-блок». Збережіть два отриманих файли: аудіофайл (.wav) та файл ключа (.key).

2. Витягування даних (Extraction).

Цей режим призначений для отримання прихованої інформації з аудіофайлу. Зовнішній вигляд вкладки наведено на рисунку 3.13.

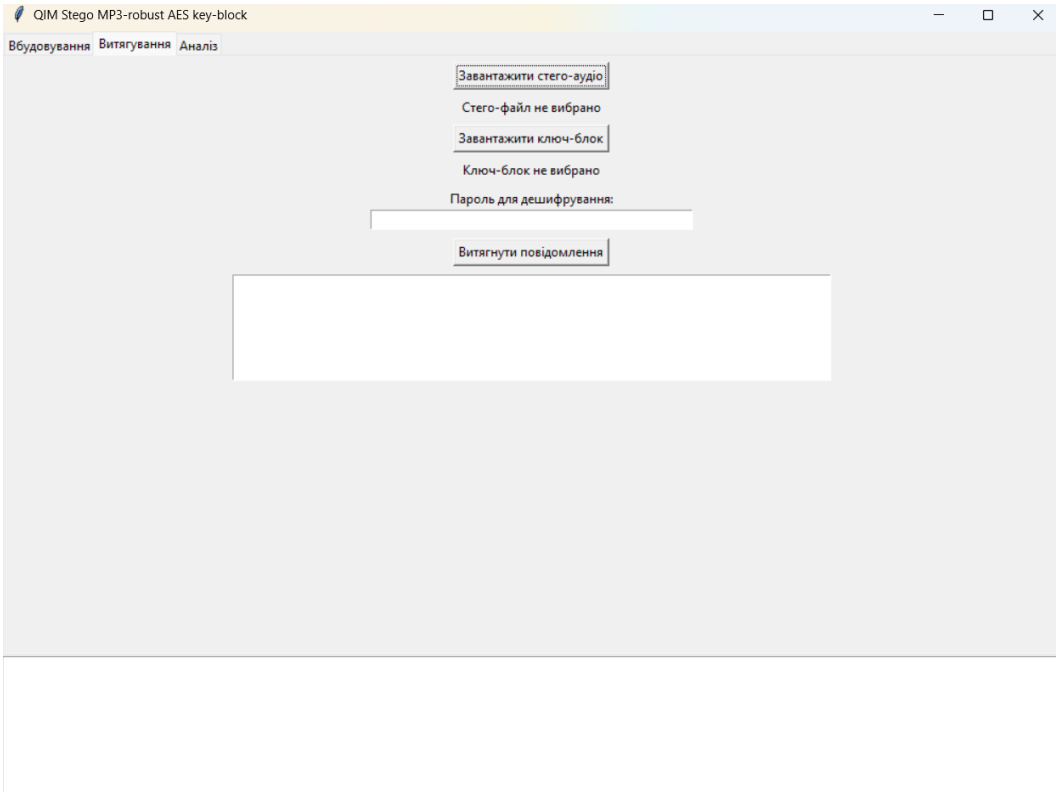


Рисунок 3.13 – Інтерфейс вкладки «Витягування»

Алгоритм дій користувача. Перейдіть на вкладку «Витягування». Натисніть «Завантажити стего-аудіо» та оберіть файл, отриманий на попередньому етапі (або файл, що пройшов MP3-стиснення). Натисніть «Завантажити ключ-блок» та оберіть відповідний файл ключа.

У полі «Пароль для дешифрування» введіть той самий пароль, що використовувався при вбудовуванні.

Важливо: Якщо пароль буде введено невірно, програма видасть помилку дешифрування, і дані не будуть вилучені.

Натисніть кнопку «Витягнути повідомлення». Результат з'явиться у великому текстовому полі. Нижче відобразиться метрика BER (Bit Error Rate). Значення BER близьке до 0.00 свідчить про успішне відновлення, високі значення (> 0.15) свідчать про пошкодження контейнера.

3. Аналіз сигналів.

Вкладка аналізу дозволяє користувачеві візуально оцінити якість роботи алгоритму та переконатися у відсутності явних артефактів. Інтерфейс режиму аналізу показано на рисунку 3.14.

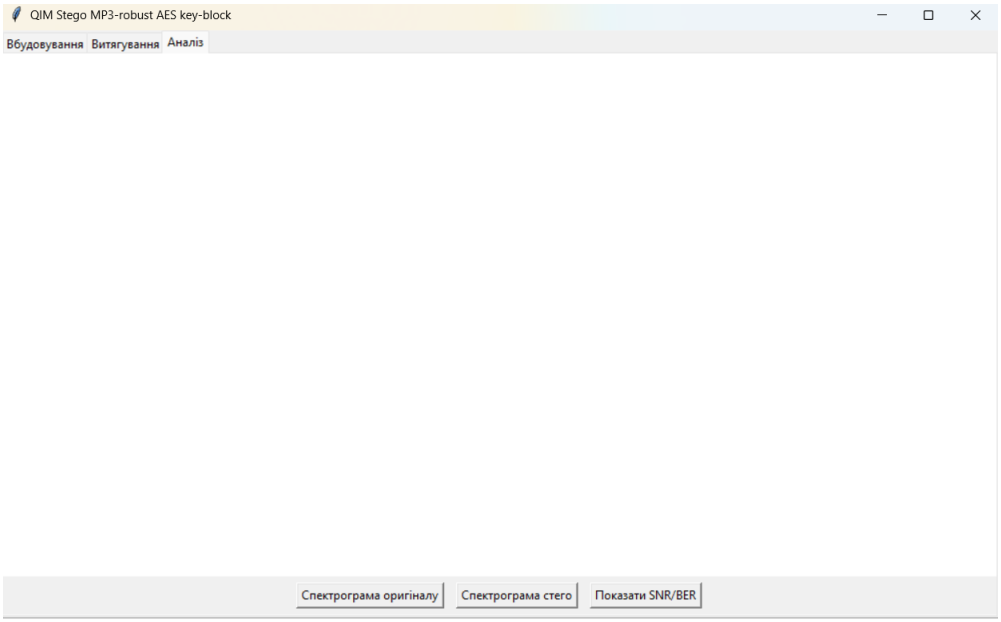


Рисунок 3.14 – Інтерфейс вкладки «Аналіз» зі спектрограмою

Функціонал вкладки. Кнопка «Спектрограма оригіналу»: будує спектральне зображення вихідного файлу. Кнопка «Спектрограма стего»: будує спектрограму файлу з прихованими даними. Користувач може порівняти ці два зображення; ідеальним результатом є їх візуальна ідентичність.

Кнопка «Показати SNR/BER»: розраховує та виводить у лог значення співвідношення сигнал/шум (SNR). Рекомендоване значення SNR для якісної стеганографії – вище 25 дБ.

Використання даної інструкції дозволяє користувачеві коректно налаштувати програмний засіб, уникнути типових помилок при введенні паролів та файлів ключів, а також самостійно оцінити надійність прихованого каналу передачі даних.

3.4 Висновки до розділу

У третьому розділі здійснено практичну реалізацію та всебічне тестування розробленого методу підвищення стійкості приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування у частотній області та шифрування ключ-блоку. Виконані роботи дозволили перейти від теоретичної моделі до діючого програмного продукту, підтвердивши ефективність запропонованих алгоритмічних рішень.

На основі обґрунтованого вибору інструментальних засобів (мова програмування Python з використанням бібліотек NumPy, SciPy, Librosa, Cryptography) розроблено архітектуру програмного забезпечення, що базується на модульному принципі. Реалізовано ключові класи: QIMQuantizer для виконання адаптивного квантування та AudioSteganographyApp для забезпечення логіки взаємодії з користувачем. Використання модифікованого дискретного косинусного перетворення (MDCT) типу IV дозволило працювати в тій же спектральній області, що й кодеки MP3, забезпечуючи сумісність та передбачуваність перетворень сигналу.

Проведене експериментальне тестування стало ключовим етапом перевірки гіпотези дослідження. Результати показали, що розроблений метод забезпечує високу стійкість прихованих даних до стиснення у формат MP3. При найбільш поширеному бітрейті 128 кбіт/с середній показник помилок бітів (BER) склав менше 0.05, а при використанні мажоритарного кодування з коефіцієнтом повторення $R=3$ вдалося досягти практично безпомилкового відновлення текстових повідомлень (BER < 0.01). Це суттєво перевищує показники класичних методів (LSB, фазове кодування), які демонструють втрату понад 80% інформації

за аналогічних умов. Стійкість методу зберігається аж до бітрейту 96 кбіт/с, після чого деградація сигналу стає критичною для коректної роботи QIM-детектора.

Аналіз якості стего-сигналів за допомогою метрики SNR (співвідношення сигнал/шум) та візуального порівняння спектрограм підтвердив високу скритність вбудовування.

У програмному засобі успішно імплементовано механізм захисту параметрів вбудовування. Ключ-блок, що містить критично важливі метадані (позиції коефіцієнтів, кроки квантування), шифрується алгоритмом симетричного блокового шифрування AES-256 у режимі CBC.

Розроблений графічний інтерфейс користувача (GUI) на базі бібліотеки Tkinter забезпечує інтуїтивно зрозуміле керування процесами вбудовування, витягування та аналізу. Розроблена інструкція користувача детально описує алгоритм роботи з програмою.

4 ЕКОНОМІЧНА ЧАСТИНА

У цьому розділі досліджено економічний потенціал розробки за темою «Підвищення стійкості методу приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування у частотній області та шифрування ключ-блоку». Аналіз включає оцінку комерційних можливостей, прогнозування витрат на виконання науково-дослідної роботи, впровадження результатів, а також оцінку очікуваних економічних вигод від реалізації розробленого програмного засобу.

Додатково проведено розрахунок ефективності вкладених інвестицій і терміну їх окупності, що є ключовими показниками для залучення потенційних інвесторів.

На основі отриманих даних буде зроблено висновок щодо економічної доцільності розробки методу стеганографічного захисту даних, що базується на сучасних алгоритмах цифрової обробки сигналів та криптографії, та її перспективності для впровадження у практичну діяльність.

4.1 Оцінювання комерційного потенціалу розробки програмного забезпечення

Метою проведення технологічного аудиту є оцінка комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

У межах магістерської роботи було розроблено програмний засіб для підвищення стійкості приховування даних в аудіосигналах. Система базується на гібридному використанні методу QIM-квантування у частотній області MDCT та криптографічного захисту параметрів вбудовування (ключ-блоку) алгоритмом AES-256. Це дозволяє забезпечити стійкість прихованої інформації до стиснення з втратами (зокрема, формату MP3), що є ключовою перевагою над існуючими аналогами.

Для проведення технологічного аудиту залучено трьох незалежних експертів. У рамках цієї роботи експертами виступають викладачі кафедри МБІС, зокрема: − Яремчук Ю. Є. (д.т.н., професор МБІС ВНТУ); − Грицак А. В. (к.т.н., доцент,

викладач кафедри МБІС ВНТУ); − Карпінець В. В. (к.т.н., доцент зав. кафедри МБІС ВНТУ).

Для оцінювання використано критерії, наведені у таблиці 4.1.

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї

Продовження таблиці 4.1

9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки зведено до таблиці 4.2.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	5	4
2. Ринкові переваги (наявність аналогів)	3	3	3
3. Ринкові переваги (ціна продукту)	5	5	5
4. Ринкові переваги (технічні властивості)	4	5	4

Продовження таблиці 4.2

5. Ринкові переваги (експлуатаційні витрати)	5	4	5
6. Ринкові перспективи (розмір ринку)	3	4	4
7. Ринкові перспективи (конкуренція)	3	3	3
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	4	4	4
10. Практична здійсненність (необхідність нових матеріалів)	5	4	5
11. Практична здійсненність (термін реалізації)	3	2	3
12. Практична здійсненність (розробка документів)	3	2	2
Сума балів	46	45	46
Середньоарифметична сума балів $СБ_с$	45,6		

На основі даних, наведених у таблиці 4.2, можна здійснити аналіз комерційного потенціалу розробки. Далі порівняємо ці результати з рівнями комерційного потенціалу, представленими в таблиці 4.3.

Таблиця 4.3 – Науково технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Результати експертного оцінювання показали, що середньоарифметична сума балів становить 45,6 балів. Це підтверджує високий науково-технічний рівень та потенційну комерційну успішність проведених досліджень, як вказано у таблиці 4.3. Отриманий високий бал зумовлений значною конкурентною перевагою та низькими експлуатаційними витратами програмного продукту.

4.2 Прогнозування витрат на виконання наукової роботи та впровадження її результатів

Під час планування, обліку та калькулювання витрат, пов'язаних із проведенням науково-дослідної роботи на тему «Підвищення стійкості методу

приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування у частотній області та шифрування ключ-блоку», витрати групуються за відповідними категоріями.

До категорії «Витрати на оплату праці» включаються витрати, пов'язані з виплатою основної та додаткової заробітної плати працівникам, які займають керівні посади у відділах, лабораторіях, секторах, групах, а також науковим, інженерно-технічним працівникам та іншим співробітникам, безпосередньо залученим до виконання цієї роботи [32].

Для визначення фонду основної заробітної плати ($З_о$) використовується аналіз трудомісткості, наведений у таблиці 4.3 (див. попередній розділ). Оскільки роботи мають дослідницький характер і виконуються частину робочого дня, розрахунок є більш точним при використанні годинних тарифних ставок.

Витрати на основну заробітну плату дослідників $З_о$ розраховуємо за формулою:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} * t_i}{T_p} \quad (4.1)$$

де k – кількість виконавців, залучених до процесу досліджень;

$M_{пі}$ – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дні;

$T_р$ – середнє число робочих днів в місяці, $T_р$= 21 дня.

$$З_о = \frac{24000}{21} \times 5 = 5714{,}3 \text{ грн.}$$

Таблиця 4.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	24000	1142,8	5	5714,3
Інженер-розробник	21000	1000	42	42000
			Всього	47714,3

Додаткову заробітну плату розраховуємо як 10 - 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \times \frac{Н_{дод}}{100\%} \quad (4.2)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати.

$Н_{дод}$ - приймемо, як 12%.

$$З_{дод} = (47714{,}3) \times \frac{12}{100\%} = 5725{,}7 \text{ грн.}$$

До статті "Відрахування на соціальні заходи" включаються внески на загальнообов'язкове державне соціальне страхування та витрати на соціальний захист населення, зокрема єдиний соціальний внесок (ЄСВ) [33].

Нарахування на заробітну плату дослідників та працівників становить 22% від суми їх основної та додаткової заробітної плати і розраховується за наступною формулою:

$$З_н = (З_о + З_р + З_{дод}) \times \frac{Н_{зп}}{100\%} \quad (4.3)$$

де $Н_{зп}$ – норма нарахування на заробітну плату.

$$З_н = (47714{,}3 + 5727{,}7) \times \frac{22}{100\%} = 11757{,}24 \text{ грн.}$$

До статті «Сировина та матеріали» відносяться витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, придбані у сторонніх підприємств, установ і організацій та використані для проведення досліджень за прямим призначенням згідно з нормами їх витрачання. Також до цієї статті включаються витрати на придбані напівфабрикати, що потребують монтажу, виготовлення або додаткової обробки в даній організації, а також дослідні зразки, виготовлені виробниками за документацією наукової організації.

Вартість матеріалів (М) розраховується окремо для кожного виду матеріалів за наступною формулою:

$$М = \sum_{j=1}^{n} H_j \times Ц_j \times K_j - \sum_{j=1}^{n} B_j \times Ц_{вj} \quad (4.4)$$

де $Н_j$ – норма витрат матеріалу j-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу j-го найменування, грн/кг;

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

$В_j$ – маса відходів j-го найменування, кг;

$Ц_{вj}$ – вартість відходів j-го найменування, грн/кг.

$$M_1 = 200 \times 2 \times 1{,}1 = 440 \text{ грн}$$

Таблиця 4.5 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за од, грн	Норма витрат, од	Вартість витраченого матеріалу, грн
Папір для принтера	200	2	440
Нотатки (стікери)	120	1	132
Канцелярський набір (ручка, олівець, лінійка)	100	2	220
Файли	70	1	77
		Всього	869

Витрати на комплектуючі (Кв), які могли б використовуватися під час проведення науково-дослідної роботи за темою «Підвищення стійкості методу приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування у частотній області та шифрування ключ-блоку», не передбачені.

До статті «Спеціальне обладнання для наукових (експериментальних) робіт» входять витрати на виготовлення та придбання спеціалізованого обладнання, яке може бути необхідним для проведення досліджень, а також витрати на його проектування, транспортування, монтаж і встановлення. У рамках цієї роботи витрати на спеціальне обладнання також не заплановані.

До статті «Програмне забезпечення для наукових (експериментальних) робіт» відносяться витрати на розробку та придбання програмного забезпечення, зокрема програм, алгоритмів і баз даних, необхідних для виконання досліджень, а також витрати на їх проектування, створення та інсталяцію. Балансова вартість програмного забезпечення розраховується за формулою:

$$В_{прг} = \sum_{i=1}^{k} Ц_{іпрг} \times С_{прг.і} \times К_{і} \quad (4.5)$$

де $Ц_{іпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$С_{прг.і}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$К_{і}$ – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($К_{і}$ = 1,10…1,12);

k – кількість найменувань програмних засобів.

$$В_{прг} = 6400 \times 2 \times 1{,}1 = 14080 \text{ грн.}$$

Таблиця 4.6 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
ОС Windows 11	2	6400	14080
GitHub CI/CD	1	5200	5720
		Всього	19800

До статті «Амортизація обладнання, програмних засобів та приміщень» включаються амортизаційні відрахування за кожним видом обладнання, устаткування, інших приладів і пристроїв, а також програмного забезпечення, які використовуються для проведення науково-дослідної роботи, за їх наявності в дослідницькій організації або на підприємстві.

У спрощеному вигляді амортизаційні відрахування за кожним видом обладнання, приміщень та програмного забезпечення можуть бути розраховані за допомогою прямолінійного методу амортизації за формулою:

$$А_{обл} = \frac{Ц_б}{Т_в} \times \frac{t_{вик}}{12} \quad (4.6)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$Т_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$А_{обл} = \frac{30000 \times 2}{2 \times 12} = 2500 \text{ грн.}$$

Таблиця 4.7 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Ноутбук LENOVO Ideapad	25000	2	2	2083,3
Ноутбук ASUS Vivobook	30000	2	2	2500
Приміщення	145000	20	2	1208,3
			Всього	5791,6

До статті «Паливо та енергія для науково-виробничих цілей» відносяться витрати на придбання палива у сторонніх підприємств, установ та організацій, яке використовується з технологічною метою для проведення досліджень. Ця стаття формується у разі проведення енергоємних наукових досліджень за методом прямого віднесення витрат і може становити значну частку у собівартості досліджень. Витрати на силову електроенергію (Ве) розраховуються за формулою:

$$В_е = \sum_{i=1}^{n} \frac{W_{yi} \times t_i \times Ц_е \times К_{впі}}{\eta_i} \quad (4.7)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_е$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_е$ = 12,50 грн;

$К_{впі}$ – коефіцієнт, що враховує використання потужності, Квпі <1;

η_i – коефіцієнт корисної дії обладнання, η_i<1.

$$В_е = \frac{0{,}4 \times 400 \times 12{,}5 \times 0{,}95}{0{,}97} = 1958{,}7 \text{ грн}$$

Таблиця 4.9 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Ноутбук LENOVO Ideapad	0,4	400	1958,7
Ноутбук ASUS Vivobook	0,4	390	1909,8
Робоче місце	0,2	360	900
		Всього	4768,5

Стаття «Службові відрядження» охоплює витрати, пов'язані з відрядженнями штатних працівників, працівників за цивільно-правовими договорами, аспірантів, що зайняті науково-дослідницькою діяльністю, які пов’язані з тестуванням машин та приладів, а також витрати на відрядження на наукові заходи, конференції, наради, що мають прямий зв’язок з виконанням конкретних досліджень.

Витрати за цією статтею розраховуються у розмірі 20–25% від суми основної заробітної плати дослідників та робітників за допомогою формули:

$$В_{сп} = (З_о + З_р) \times \frac{Н_{сп}}{100\%} \quad (4.8)$$

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо $Н_{сп}$= 30%.

$$В_{сп} = 47714{,}3 \times \frac{30\%}{100\%} = 148314{,}3 \text{ грн.}$$

Стаття «Інші витрати» включає витрати, які не були охарактеризовані у попередніх статтях витрат і можуть бути прямо віднесені до собівартості досліджень за безпосередніми показниками. Витрати за цією статтею обчислюються у розмірі 50–100% від суми основної заробітної плати дослідників та робітників за допомогою такої формули:

$$I_{\text{ів}} = (З_{\text{о}} + З_{\text{р}}) \times \frac{Н_{\text{ів}}}{100\%} \quad (4.9)$$

де $Н_{\text{ів}}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{\text{ів}} = 50\%$.

$$I_{\text{ів}} = 47714{,}3 \times \frac{50}{100} = 23857{,}2 \text{ грн.}$$

Сталими (загальновиробничими) витратами охоплюються різноманітні витрати, пов'язані з управлінням організацією, зусиллями в інноваціях та раціоналізації, а також з набором та підготовкою персоналу, банківськими послугами, освоєнням виробництва, а також науково-технічною інформацією та рекламою.

Витрати за цією статтею розраховуються у розмірі 100–150% від суми основної заробітної плати дослідників та працівників з використанням такої формули:

$$В_{\text{нзв}} = (З_{\text{о}} + З_{\text{р}}) \times \frac{Н_{\text{нзв}}}{100\%} \quad (4.10)$$

де $Н_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{\text{нзв}} = 100\%$.

$$В_{\text{нзв}} = 47714{,}3 \times \frac{100}{100} = 47714{,}3 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$В_{\text{заг}} = З_{\text{о}} + З_{\text{р}} + З_{\text{дод}} + З_{\text{н}} + М + К_{\text{в}} + В_{\text{спец}} + В_{\text{прг}} + А_{\text{обл}} + В_{\text{е}} + В_{\text{св}} + В_{\text{сп}} + І_{\text{в}} + В_{\text{нзв}} \quad (4.11)$$

$$В_{\text{заг}} = 47714{,}3 + 5725{,}7 + 11757{,}24 + 869 + 14080 + 2500 + 1958{,}7 + 148314{,}3 + 23857{,}2 + 47714{,}3 = 304490{,}74 \text{ грн.}$$

Вартість завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів обчислюється відповідно до наступної формули:

$$ЗВ = \frac{В_{заг}}{\eta} \quad (4.12)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науководослідної роботи, приймемо η =0,7.

$$ЗВ = \frac{304490,74}{0,7} = 434986,8 \text{ грн.}$$

Отже, прогноз загальних витрат ЗВ на виконання та впровадження результатів виконаної роботи складає 434986,8 грн.

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

У ринкових умовах позитивний результат від можливого впровадження науково-технічної розробки для потенційного інвестора полягає у збільшенні чистого прибутку. Дослідження з підвищення стійкості методу приховування даних в аудіосигналах передбачають комерціалізацію протягом трьох років.

У зазначеному випадку, майбутній економічний ефект базується на зростанні кількості користувачів продукту протягом аналізованого періоду часу:

у перший рік – 180 користувачів;

у другий – 220 користувачів;

у третій – 200 користувачів.

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, приймемо 2000 користувачів;

$Ц_б$ – вартість програмного продукту у році до впровадження результатів розробки, приймемо 20000,00 грн;

$\pm\Delta Ц_о$ – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, приймемо 1200,00 грн.

Для кожного з випадків потенційне збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ в роки очікуваного позитивного результату від можливого впровадження та комерціалізації науково-технічної розробки розраховується за відповідною формулою:

$$\Delta П_i = (\pm \Delta Ц_o \times N + Ц_o \times N_i) \times \lambda \times \rho \times (1 - \frac{\vartheta}{100}) \qquad (4.14)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2025 році ставка податку на додану вартість складає 20%, а коефіцієнт λ =0,8333;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту. Приймемо ρ =30%;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2025 році ϑ =18%;

Збільшення чистого прибутку 1-го року:

$$\Delta П_1 = (1200 \times 2000 \times 20000 \times 180) \times 0{,}83 \times 0{,}3 \times \left(1 - \frac{0{,}18}{100}\right)$$

$$= 2\ 147\ 487{,}6 \text{ грн}$$

Збільшення чистого прибутку 2-го року:

$$\Delta П_2 = (1200 \times 2000 \times 20000 \times (180 + 220)) \times 0{,}83 \times 0{,}3 \times \left(1 - \frac{0{,}18}{100}\right)$$

$$= 4\ 772\ 194{,}6 \text{ грн}$$

Збільшення чистого прибутку 3-го року:

$$\Delta П_1 = (1200 \times 2000 \times 20000 \times (180 + 220 + 200)) \times 0{,}83 \times 0{,}3 \times \left(1 - \frac{0{,}18}{100}\right)$$

$$= 7\ 158\ 291{,}8 \text{ грн}$$

Для кожного з випадків потенційне збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ в роки очікуваного позитивного результату від можливого впровадження та комерціалізації науково-технічної розробки розраховується за відповідною формулою:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t} \quad (4.15)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

Т – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ =0,2;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = \frac{2\ 147\ 487{,}6}{(1+0{,}2)^1} + \frac{4\ 772\ 194{,}6}{(1+0{,}2)^2} + \frac{7\ 158\ 291{,}8}{(1+0{,}2)^3} = 9\ 246\ 127 \text{ грн.}$$

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Ключовими факторами, що визначають обґрунтованість інвестування певним інвестором у наукову розробку, є абсолютна та відносна ефективність інвестицій, а також термін їх повернення. Першим кроком на цьому шляху є розрахунок сучасної вартості інвестицій (PV), вкладених у наукову розробку.

Для цього можна використати формулу:

$$PV = k_{інв} \times ЗВ \quad (4.16)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науковотехнічної розробки та її комерціалізацію, приймаємо $k_{інв}$ =3;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 434986,8 грн.

$$PV = 3 \times 434986{,}8 = 1\ 304\ 960{,}4 \text{ грн.}$$

Таким чином, чистий приведений дохід (NPV) або абсолютний економічний ефект ($Е_{абс}$) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки буде таким:

$$Е_{абс} = ПП - PV \quad (4.17)$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 9 246 127 грн;

PV – теперішня вартість початкових інвестицій, 1 304 960,4 грн.

$$Е_{абс} = 9\,246\,127 - 1\,304\,960{,}4 = 7\,941\,166{,}6 \text{ грн.}$$

Внутрішня економічна дохідність (Ев) інвестицій, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, обчислюється за допомогою такої формули:

$$Е_в = \sqrt[Т_ж]{1 + \frac{Е_{абс}}{PV}}, \quad (4.18)$$

де $Е_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 8 210 408,2 грн;

PV – теперішня вартість початкових інвестицій, 1 035 718,8 грн;

$Т_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 3 роки.

$$Ев = \sqrt[3]{1 + \frac{7\,941\,166{,}6}{1\,304\,960{,}4}} - 1 = 0{,}7$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій (мін τ) визначається згідно такою формулою:

$$\tau_{мін} = d + f, \quad (4.19)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2025 році в Україні d =0,15;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,2.

$$\tau_{min} = 0{,}2 + 0{,}15 = 0{,}35$$

Оскільки Ев = 70% > τmin = 35%, це свідчить про те, що внутрішня економічна дохідність інвестицій, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, перевищує мінімальну внутрішню дохідність. Таким чином, інвестування у науково-дослідну роботу за темою «Підвищення стійкості методу приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування у частотній області та шифрування ключ-блоку» є економічно обґрунтованим і доцільним.

Далі обчислюємо період окупності інвестицій ($T_{ок}$ або DPP, Discounted Payback Period), які потенційний інвестор може вкласти у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}\ , \tag{4.20}$$

$$T_{ок} = \frac{1}{0{,}7} = \ 1{,}5 \text{ року}.$$

З огляду на те, що період окупності інвестицій у реалізацію наукового проєкту становить менше трьох років, можна дійти висновку, що фінансування цієї нової розробки є виправданим.

4.5 Висновки до розділу

У ході виконання економічного аналізу було встановлено, що розроблений програмний засіб має високий комерційний потенціал. За результатами експертного оцінювання середньоарифметичний показник становить 45,6 бала, що відповідає категорії «високий рівень» і свідчить про конкурентоспроможність технології на ринку та реальну можливість її комерціалізації.

Прогнозовані витрати на розробку та впровадження становлять 434 986,8 грн, а теперішня вартість очікуваних інвестицій — 1 304 960,4 грн. При цьому приведена вартість майбутніх прибутків оцінюється у 9 246 127 грн, що забезпечує значний позитивний чистий приведений дохід. Такий результат однозначно демонструє економічну доцільність розробки.

Внутрішня економічна дохідність проєкту становить 70%, що перевищує мінімально допустиме значення 35%, розраховане з урахуванням депозитної ставки та ризиковості інвестицій. Крім того, період окупності не перевищує трьох років, що також підтверджує привабливість інвестування у впровадження розробленого методу й програмного забезпечення на його основі.

Отримані результати узгоджуються між собою й показують, що розробка має не лише технічну і наукову цінність, а й реальні перспективи практичного застосування. Вона здатна забезпечити істотний економічний ефект, швидку окупність і конкурентні переваги на ринку рішень у сфері цифрової безпеки.

Таким чином, проведення науково-дослідної роботи є повністю обґрунтованим, економічно доцільним і перспективним з точки зору подальшої комерціалізації та впровадження у практичну діяльність.

ВИСНОВОК

У ході виконання магістерської роботи було розроблено та впровадженно метод підвищення стійкості приховування інформації в аудіосигналах до стиснення MP3, що поєднує QIM-квантування в частотній області MDCT та криптографічний захист параметрів вбудовування з використанням AES-256. Основною метою дослідження було створення надійного, практично застосовного та захищеного стеганографічного методу, здатного забезпечувати передачу прихованих даних у середовищах із втратним кодуванням без суттєвих спотворень та ризику втрати корисної інформації.

У межах роботи проведено ґрунтовний аналітичний огляд сучасних стеганографічних підходів, особливо тих, що орієнтовані на аудіоканали з втратами. Було визначено, що більшість класичних методів LSB, Spread Spectrum, Echo Hiding тощо. мають низьку стійкість до MP3-стиснення через агресивну перцептивну фільтрацію. Це підтвердило актуальність вибраної наукової проблеми та визначило напрямок удосконалення методів приховування.

Розроблений метод базується на кількісній модифікації спектральних коефіцієнтів MDCT, яка демонструє природну стійкість до втратної компресії, а також на шифруванні службового ключ-блоку, що містить параметри вбудовування. Такий гібридний підхід дозволив сформувати систему, яка поєднує переваги стеганографії та криптографії — приховування факту передачі даних та надійний захист структури прихованого каналу.

Програмна реалізація включала розробку повноцінного програмного засобу з інтуїтивним графічним інтерфейсом. Було створено модулі вбудовування, вилучення, шифрування/дешифрування ключ-блоку, аналізу сигналів, а також окрему інструкцію користувача. Проведене тестування на різних аудіосигналах та бітрейтах MP3 підтвердило працездатність розробленого методу.

Отримані результати довели, що розроблений метод здатний ефективно працювати в умовах реальних аудіосистем. Він демонструє високу стійкість до MP3-стиснення, не створює помітних артефактів в аудіосигналі та гарантує захист

службових параметрів завдяки криптографії. Це робить метод придатним для практичного застосування в системах передавання конфіденційної інформації, мультимедійних сервісах, сховищах даних та спеціалізованих засобах зв’язку.

Результати роботи можуть бути використані в корпоративних системах захисту інформації, у стеганографічних комплексах для безпечного обміну даними, а також у наукових дослідженнях, що стосуються прихованих каналів зв’язку. Подальший розвиток може включати розширення методу для багатоканальних форматів, оптимізацію під інші види компресії AAC, OGG, впровадження адаптивного квантування та використання глибоких нейронних мереж для підвищення непомітності та стійкості.

Таким чином, усі поставлені завдання були повністю виконані, а отримані результати підтвердили доцільність застосування запропонованих підходів. Розробка довела свою ефективність як з технічної, так і з економічної точки зору.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Petitcolas F. A. P., Anderson R. J., Kuhn M. G. Information hiding-a survey. Proceedings of the IEEE. 1999. Т. 87, № 7. С. 1062–1078. URL: https://doi.org/10.1109/5.771065 (дата звернення: 09.09.2025).

2. Chen B., Wornell G. W. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Transactions on Information Theory. 2001. Т. 47, № 4. С. 1423–1443. URL: https://doi.org/10.1109/18.923725 (дата звернення: 10.09.2025).

3. Abdallah H. A., Meshoul S. A Multilayered Audio Signal Encryption Approach for Secure Voice Communication. Electronics. 2022. Т. 12, № 1. С. 2. URL: https://doi.org/10.3390/electronics12010002 (дата звернення: 09.09.2025).

4. Techniques for data hiding / W. Bender та ін. IBM Systems Journal. 1996. Т. 35, № 3.4. С. 313–336. URL: https://doi.org/10.1147/sj.353.0313 (дата звернення: 12.09.2025).

5. Society A. E. AES 2024 Acoustics & Sound Reinforcement Conference. Audio Engineering Society, 2024.

6. Petitcolas F. A. P., Anderson R. J., Kuhn M. G. Information hiding-a survey. Proceedings of the IEEE. 1999. Т. 87, № 7. С. 1062–1078. URL: https://doi.org/10.1109/5.771065 (дата звернення: 14.09.2025).

7. Fridrich J. Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge University Press, 2014.

8. Image Steganography Using LSB and Hybrid Encryption Algorithms / M. Alanzy та ін. Applied Sciences. 2023. Т. 13, № 21. С. 11771. URL: https://doi.org/10.3390/app132111771 (дата звернення: 15.09.2025).

9. An Adaptive Steganographic Method for Reversible Information Embedding in X-Ray Images / E. Daiyrbayeva та ін. Computers. 2025. Т. 14, № 9. С. 386. URL: https://doi.org/10.3390/computers14090386 (дата звернення: 18.09.2025).

10. Petitcolas F. A. P., Anderson R. J., Kuhn M. G. Information hiding-a survey. *Proceedings of the IEEE*. 1999. Т. 87, № 7. С. 1062–1078. URL: https://doi.org/10.1109/5.771065 (дата звернення: 23.09.2025).

11. Schneier B. Applied Cryptography: Protocols, Algorithms and Source Code in C. Wiley & Sons, Incorporated, John, 2017. 784 p.

12. Painter T., Spanias A. Perceptual coding of digital audio. *Proceedings of the IEEE*. 2000. Т. 88, № 4. С. 451–515. URL: https://doi.org/10.1109/5.842996 (дата звернення: 25.09.2025).

13. Handbook of applied cryptography. *Choice Reviews Online*. 1997. Т. 34, № 08. С. 34–4512–34–4512. URL: https://doi.org/10.5860/choice.34-4512 (дата звернення: 27.09.2025).

14. Daemen J., Rijmen V. Design of Rijndael: AES - the Advanced Encryption Standard. Springer London, Limited, 2013. 238 p.

15. Signal processing with lapped transforms. Boston : Artech House, 1991. 357 p.

16. Fridrich J., Goljan M., Rui Du. Detecting LSB steganography in color, and gray-scale images. *IEEE Multimedia*. 2001. Т. 8, № 4. С. 22–28. URL: https://doi.org/10.1109/93.959097 (дата звернення: 30.09.2025).

17. Tanenbaum A. S., Wetherall D. Computer Networks, EBook, Global Edition. Pearson Education, Limited, 2021.

18. Cormen T. H., Stein C., Rivest R. L. Introduction to Algorithms, Fourth Edition. MIT Press, 2022. 1332 p.

19. Sinha D., Tewfik A. H. Low bit rate transparent audio compression using adapted wavelets. *IEEE Transactions on Signal Processing*. 1993. Т. 41, № 12. С. 3463–3479. URL: https://doi.org/10.1109/78.258086 (дата звернення: 04.10.2025).

20. Wayne K., Sedgewick R. Algorithms. Pearson Education, Limited, 2010. 432 p.

21. Kozyrakis C., Hennessy J. L., Patterson D. A. Computer Architecture: A Quantitative Approach. Elsevier Science & Technology Books, 2024.

22. McKinney W. Python for Data Analysis. O'Reilly Media, Incorporated, 2022.

23. Python Cryptographic Authority. *Cryptography Documentation*. URL: https://cryptography.io/ (дата звернення: 06.10.2025).

24. Bastian R. PySoundFile: An audio library based on libsndfile, CFFI and NumPy. *GitHub*. URL: https://github.com/bastibe/PySoundFile (дата звернення: 08.10.2025).

25. Roy J. Pydub: Manipulate audio with a simple and easy high level interface. *GitHub*. URL: https://github.com/jiaaro/pydub (дата звернення: 10.10.2025).

26. Krekel H., Oliveira B., Pfannschmidt R. Helps you write better programs. *pytest documentation*. URL: https://docs.pytest.org/ (дата звернення: 14.10.2025).

27. NumPy Documentation. *NumPy*. URL: https://numpy.org/doc (дата звернення: 15.10.2025).

28. Numpy and Scipy Documentation. *Scipy.org*. URL: https://docs.scipy.org/doc (дата звернення: 16.10.2025).

29. Librosa Documentation. *Librosa*. URL: https://librosa.org/doc (дата звернення: 17.10.2025).

30. Cryptography documentation. *Cryptography*. URL: https://cryptography.io/ (дата звернення: 20.10.2025).

31. Matplotlib documentation. *Matplotlib*. URL: https://matplotlib.org/stable (дата звернення: 23.10.2025).

32. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

33. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа. Вінниця : ВНТУ, 2016. 113 с

34. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 125 «Кібербезпека» (освітня програма «Кібербезпека інформаційних технологій та систем») [Електонний ресурс] / уклад.: Ю. Є. Яремчук, В. В. Карпінець. − Вінниця : ВНТУ, 2023. – 43 с.

ДОДАТКИ

Додаток А. Технічне завдання

Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ

Голова секції "Управління інформаційною безпекою" кафедри МБІС
д.т.н., професор

_____ Юрій ЯРЕМЧУК
"_22_" __вересня_____ 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:

«Підвищення стійкості методу приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування у частотній області та шифрування ключ-блоку»

08-72.МКР.017.00.086.ТЗ

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

Вінниця – 2025 р.

1. Найменування та область застосування

Програмний засіб підвищення стійкості приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування у частотній області та шифрування ключ-блоку. Область застосування: системи кібербезпеки, приховані канали передавання службових даних, захист інформації в мультимедійних системах.

2.Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ №96 від 20. 03. 2025 р.

3. Мета та призначення розробки

3.1 Мета розробки: розробка методу та програмного засобу для стійкого приховування даних в аудіосигналах, який зберігає працездатність після MP3-компресії, шляхом використання QIM-квантування у частотній області та шифрування ключ-блоку.

3.2 Призначення: вбудовування прихованих даних у аудіосигнал, забезпечення стійкості прихованих повідомлень до стиснення MP3, захисту ключових параметрів вбудовування шляхом криптографічного шифрування.

4. Джерела розробки

4.1. Petitcolas F. A. P., Anderson R. J., Kuhn M. G. Information hiding-a survey. Proceedings of the IEEE. 1999. Т. 87, № 7. С. 1062–1078.

4.2. Chen B., Wornell G. W. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. IEEE Transactions on Information Theory. 2001. Т. 47, № 4. С. 1423–1443.

4.3. Abdallah H. A., Meshoul S. A Multilayered Audio Signal Encryption Approach for Secure Voice Communication. Electronics. 2022. Т. 12, № 1. С. 2.

4.4. Society A. E. AES 2024 Acoustics & Sound Reinforcement Conference. Audio Engineering Society, 2024.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний, легкий у використанні інтерфейс користувача;

5.1.2 Реалізація методу не повинна вимагати спеціальних ліцензійних програмних додатків;

5.1.3 Програмний засіб повинен виконувати процес приховування даних в аудіосигналах.

5.2 Вимоги до надійності:

5.2.1 Програма повинна забезпечувати коректну роботу при некоректному введенні даних та виводити відповідні діагностичні повідомлення;

5.2.2 Має бути забезпечена перевірка цілісності ключ-блоку після дешифрування;

5.2.3 Програмний засіб повинен виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – Pentium Gold 2900 МГц і подібні до них;

– оперативна пам’ять – не менше 4 Gb;

– середовище функціонування – операційна система сімейство Windows;

– вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп’ютерною технікою.

6. Вимоги до програмної документації

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

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист розроблюваного програмного засобу від несанкціонованого використання.

7.2 Неможливість отримання доступу незареєстрованих користувачів до інформаційних ресурсів.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

8.2 Має бути реалізований таким чином, щоб підходити для використання широкого загалу.

9. Стадії та етапи розробки

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Початок	Закінчення
1	Визначення напрямку магістерської роботи, формулювання теми		
2	Аналіз предметної області обраної теми		
3	Апробація отриманих результатів		
4	Розробка алгоритму роботи		
5	Написання магістерської роботи на основі розробленої теми		
6	Розробка економічної частини		
7	Передзахист магістерської кваліфікаційної роботи		
8	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи		
9	Захист магістерської кваліфікаційної роботи		

10. Порядок контролю та прийому

10.1 До приймання магістерської кваліфікаційної роботи надається:

– ПЗ до магістерської кваліфікаційної роботи;

– програмний додаток;

– презентація;

– відзив керівника роботи;

– відзив опонента.

Технічне завдання до виконання прийняв _____ Олексюк Є. М.

Додаток Б. Лістинг програми

```
import tkinter as tk
from tkinter import filedialog, messagebox, ttk
import numpy as np
import librosa
import soundfile as sf
from scipy.fft import dct, idct
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
from cryptography.hazmat.primitives import hashes, padding
from cryptography.hazmat.primitives.kdf.pbkdf2 import PBKDF2HMAC
import os, json, threading, tempfile, traceback
from pydub import AudioSegment
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
class QIMQuantizer:
    def __init__(self, delta=0.01, alpha=0.6):
        self.delta = float(delta)
        self.alpha = float(alpha)
    def quantize(self, coeff, bit):
        q0 = self.delta * np.round(coeff / self.delta)
        q1 = self.delta * np.round((coeff - self.delta / 2) / self.delta) + self.delta / 2
        q_target = q1 if int(bit) else q0
        return float(coeff + (q_target - coeff) * self.alpha)
    def dequantize(self, coeff):
        q0 = self.delta * np.round(coeff / self.delta)
        q1 = self.delta * np.round((coeff - self.delta / 2) / self.delta) + self.delta / 2
        return 1 if abs(coeff - q1) < abs(coeff - q0) else 0
def derive_key(password: str, salt: bytes, iterations: int = 200000):
    kdf = PBKDF2HMAC(algorithm=hashes.SHA256(), length=32, salt=salt, iterations=iterations)
    return kdf.derive(password.encode())
def encrypt_bytes(plaintext_bytes: bytes, password: str):
    salt = os.urandom(16)
    key = derive_key(password, salt)
    iv = os.urandom(16)
    padder = padding.PKCS7(128).padder()
    padded = padder.update(plaintext_bytes) + padder.finalize()
    cipher = Cipher(algorithms.AES(key), modes.CBC(iv))
    encryptor = cipher.encryptor()
    ct = encryptor.update(padded) + encryptor.finalize()
    return salt + iv + ct
def decrypt_bytes(blob: bytes, password: str):
    if len(blob) < 32:
        raise ValueError("Encrypted blob too short")
    salt, iv, ct = blob[:16], blob[16:32], blob[32:]
    key = derive_key(password, salt)
    cipher = Cipher(algorithms.AES(key), modes.CBC(iv))
    decryptor = cipher.decryptor()
    padded = decryptor.update(ct) + decryptor.finalize()
    unpadder = padding.PKCS7(128).unpadder()
    data = unpadder.update(padded) + unpadder.finalize()
    return data
class AudioSteganographyApp:
    def __init__(self, root):
        self.root = root
        self.root.title("MDCT QIM Stego — MP3-robust + AES key-block")
        self.root.geometry("1000x720")
        self.audio_path = None
        self.original_audio = None
```

```
        self.stego_audio = None
        self.sr = None
        self.message = ""
        self.plain_key_block = None
        self.encrypted_key_block = None
        self._busy_lock = threading.Lock()
        self.window_size = 2048
        self.hop_size = 1024
        self.embed_band_low_frac = 0.12
        self.embed_band_high_frac = 0.35
        self.qim_alpha = 0.6
        self.qim_base_delta = 0.01
        self.redundancy = 3  # для більшої надійності
        self.notebook = ttk.Notebook(root)
        self.notebook.pack(fill=tk.BOTH, expand=True)
        self.embed_tab = ttk.Frame(self.notebook)
        self.notebook.add(self.embed_tab, text="Вбудовування")
        tk.Button(self.embed_tab, text="Завантажити аудіо (wav/mp3/...)",
command=self.load_audio).pack(pady=6)
        self.audio_label = tk.Label(self.embed_tab, text="Файл не вибрано"); self.audio_label.pack()
        tk.Label(self.embed_tab, text="Повідомлення (ASCII текст):").pack(pady=(8,0))
        self.message_entry = tk.Text(self.embed_tab, height=5, width=70); self.message_entry.pack()
        tk.Label(self.embed_tab, text="Пароль для шифру ключ-блоку:").pack(pady=(8,0))
        self.key_entry = tk.Entry(self.embed_tab, width=50, show="*"); self.key_entry.pack()
        self.encrypt_var = tk.BooleanVar(value=True)
        tk.Checkbutton(self.embed_tab, text="Шифрувати ключ-блок", variable=self.encrypt_var).pack(pady=(4,0))
        tk.Label(self.embed_tab, text="Бітрейт MP3 для симуляції:").pack(pady=(8,0))
        self.bitrate_var = tk.StringVar(value="128")
        tk.OptionMenu(self.embed_tab, self.bitrate_var, "64","96","128","192","320").pack()
        tk.Button(self.embed_tab, text="Вбудувати", command=self.start_embed).pack(pady=8)
        tk.Button(self.embed_tab, text="Вбудувати + MP3 симуляція",
command=self.start_embed_and_simulate).pack(pady=4)
        self.progress = ttk.Progressbar(self.embed_tab, orient="horizontal", mode="determinate")
        self.progress.pack(fill=tk.X, pady=6)
        tk.Button(self.embed_tab, text="Зберегти стего та ключ-блок",
command=self.save_stego_audio).pack(pady=6)
        self.extract_tab = ttk.Frame(self.notebook)
        self.notebook.add(self.extract_tab, text="Витягування")
        tk.Button(self.extract_tab, text="Завантажити стего-аудіо", command=self.load_stego_audio).pack(pady=6)
        self.stego_label = tk.Label(self.extract_tab, text="Стего-файл не вибрано"); self.stego_label.pack()
        tk.Button(self.extract_tab, text="Завантажити ключ-блок", command=self.load_key_block).pack(pady=6)
        self.key_block_label = tk.Label(self.extract_tab, text="Ключ-блок не вибрано"); self.key_block_label.pack()
        tk.Label(self.extract_tab, text="Пароль для дешифрування:").pack(pady=(6,0))
        self.extract_key_entry = tk.Entry(self.extract_tab, width=50, show="*"); self.extract_key_entry.pack()
        tk.Button(self.extract_tab, text="Витягнути повідомлення", command=self.start_extract).pack(pady=8)
        self.extracted_text = tk.Text(self.extract_tab, height=6, width=70); self.extracted_text.pack()
        self.ber_label = tk.Label(self.extract_tab, text=""); self.ber_label.pack()
        self.analysis_tab = ttk.Frame(self.notebook)
        self.notebook.add(self.analysis_tab, text="Аналіз")
        self.figure = plt.Figure(figsize=(7,4), dpi=100)
        self.canvas = FigureCanvasTkAgg(self.figure, master=self.analysis_tab)
        self.canvas.get_tk_widget().pack(fill=tk.BOTH, expand=True)
        btn_frame = ttk.Frame(self.analysis_tab); btn_frame.pack(pady=6)
        tk.Button(btn_frame, text="Спектрограма оригіналу",
command=self.plot_spectrogram_original).pack(side=tk.LEFT, padx=6)
        tk.Button(btn_frame, text="Спектрограма стего",
command=self.plot_spectrogram_stego).pack(side=tk.LEFT, padx=6)
        tk.Button(btn_frame, text="Показати SNR/BER", command=self.show_metrics).pack(side=tk.LEFT, padx=6)
```

```
    self.log_text = tk.Text(root, height=8); self.log_text.pack(fill=tk.X)
  def log(self, msg):
    try: self.log_text.insert(tk.END, msg+"\n"); self.log_text.see(tk.END)
    except: pass
  def _convert_to_wav_and_load(self, path):
    ext = os.path.splitext(path)[1].lower()
    if ext == ".wav":
      audio, sr = librosa.load(path, sr=None, mono=True)
    else:
      tmp = tempfile.NamedTemporaryFile(suffix=".wav", delete=False); tmp.close()
      audioseg = AudioSegment.from_file(path); audioseg.export(tmp.name, format="wav")
      audio, sr = librosa.load(tmp.name, sr=None, mono=True)
      os.remove(tmp.name)
    return audio, sr
  def load_audio(self):
    try:
      path = filedialog.askopenfilename(filetypes=[("Audio files","*.wav *.mp3 *.flac *.ogg *.m4a")])
      if not path: return
      self.audio_path = path
      self.original_audio, self.sr = self._convert_to_wav_and_load(path)
      self.stego_audio = self.original_audio.copy()
      self.audio_label.config(text=f"Завантажено: {os.path.basename(path)}")
      self.log("Оригінальне аудіо завантажено успішно.")
    except Exception as e: self.log("Помилка завантаження: "+str(e))
  def load_stego_audio(self):
    try:
      path = filedialog.askopenfilename(filetypes=[("Audio files","*.wav *.mp3 *.flac *.ogg *.m4a")])
      if not path: return
      self.stego_audio, self.sr = self._convert_to_wav_and_load(path)
      self.stego_label.config(text=f"Завантажено: {os.path.basename(path)}")
      self.log("Стего-аудіо завантажено успішно.")
    except Exception as e: self.log("Помилка завантаження стего: "+str(e))
  def load_key_block(self):
    path = filedialog.askopenfilename(filetypes=[("Key files","*.key *.json")])
    if not path: return
    try:
      with open(path,'rb') as f: data=f.read()
      try:
        kb = json.loads(data.decode('utf-8'))
        self.plain_key_block = kb; self.encrypted_key_block = None
        self.key_block_label.config(text=f"Завантажено (plain): {os.path.basename(path)}")
        self.log("Ключ-блок (plain) завантажено успішно.")
      except:
        self.encrypted_key_block = data; self.plain_key_block = None
        self.key_block_label.config(text=f"Завантажено (encrypted): {os.path.basename(path)}")
        self.log("Завантажено зашифрований ключ-блок.")
    except Exception as e: self.log("Помилка завантаження ключ-блоку: "+str(e))
  def mdct(self, signal, window_size=None, hop_size=None):
    if window_size is None: window_size=self.window_size
    if hop_size is None: hop_size=self.hop_size
    frames=[]
    if len(signal)<window_size: signal=np.pad(signal,(0,window_size-len(signal)))
    for i in range(0,len(signal)-window_size+1,hop_size):
      frame = signal[i:i+window_size]*np.hanning(window_size)
      frames.append(dct(frame,type=4,norm='ortho'))
    return frames
  def imdct(self, mdct_frames, window_size=None, hop_size=None):
    if window_size is None: window_size=self.window_size
    if hop_size is None: hop_size=self.hop_size
```

```
        signal=[]
        for frame in mdct_frames:
            time_frame=idct(frame,type=4,norm='ortho')*np.hanning(window_size)
            signal.extend(time_frame)
        return np.array(signal)
    def start_embed(self):
        if self._busy_lock.locked(): messagebox.showinfo("Зайнято","Інша операція виконується."); return
        threading.Thread(target=self._embed_thread,daemon=True).start()
    def start_embed_and_simulate(self):
        if self._busy_lock.locked(): messagebox.showinfo("Зайнято","Інша операція виконується."); return
        threading.Thread(target=self._embed_sim_thread,daemon=True).start()
    def _embed_thread(self):
        with self._busy_lock:
            try: self._embed_message()
            except Exception as e: self.log("Помилка embed: "+str(e)); traceback.print_exc()
    def _embed_sim_thread(self):
        with self._busy_lock:
            try:
                self._embed_message()
                self._simulate_mp3(self.bitrate_var.get())
            except Exception as e: self.log("Помилка embed+simulate: "+str(e)); traceback.print_exc()
    def _embed_message(self):
        if self.original_audio is None: messagebox.showerror("Помилка","Завантажте аудіо."); return
        self.message=self.message_entry.get("1.0",tk.END).strip()
        if not self.message: messagebox.showerror("Помилка","Введіть повідомлення."); return
        bits=''.join(format(ord(c),'08b') for c in self.message)[:128]
        bits_redundant=''.join(b*self.redundancy for b in bits)
        total_bits=len(bits_redundant)
        mdct_frames=self.mdct(self.original_audio,self.window_size,self.hop_size)
        frame=mdct_frames[0]  # один блок
        embed_positions=np.linspace(int(0.12*len(frame)),int(0.35*len(frame))-1,total_bits,dtype=int)
        key_block={"frame_idx":0,"positions":[],"deltas":[]}
        for idx,pos in enumerate(embed_positions):
            coeff=frame[pos]; delta=max(0.002,self.qim_base_delta*(1+abs(coeff)/10))
            q=QIMQuantizer(delta=delta,alpha=self.qim_alpha)
            bit=int(bits_redundant[idx])
            frame[pos]=q.quantize(coeff,bit)
            key_block["positions"].append(int(pos)); key_block["deltas"].append(float(delta))
        self.stego_audio=self.imdct(mdct_frames,self.window_size,self.hop_size)
        snr=self.calculate_snr()
        if snr<12: self.log(f"Попередження: SNR низький ({snr:.2f} dB).")
        self.log(f"Вбудовування завершено. SNR: {snr:.2f} dB")
        encrypt_kb=bool(self.encrypt_var.get()); password=self.key_entry.get().strip()
        if encrypt_kb:
            if not password: messagebox.showerror("Помилка","Пароль для шифру не вказано."); return
            blob=encrypt_bytes(json.dumps(key_block).encode('utf-8'),password)
            self.encrypted_key_block=blob; self.plain_key_block=None
        else: self.plain_key_block=key_block; self.encrypted_key_block=None
    def _simulate_mp3(self, bitrate_str='128'):
        if self.stego_audio is None: messagebox.showerror("Помилка","Спочатку вбудуйте повідомлення."); return
        try:
            tmp_wav=tempfile.NamedTemporaryFile(suffix=".wav",delete=False); tmp_wav.close()
            sf.write(tmp_wav.name,self.stego_audio,self.sr)
            tmp_mp3=tempfile.NamedTemporaryFile(suffix=".mp3",delete=False); tmp_mp3.close()

AudioSegment.from_wav(tmp_wav.name).export(tmp_mp3.name,format="mp3",bitrate=f"{bitrate_str}k")
            self.stego_audio,_=librosa.load(tmp_mp3.name,sr=self.sr,mono=True)
            os.remove(tmp_wav.name); os.remove(tmp_mp3.name)
            self.log(f"MP3 симуляція завершена ({bitrate_str} kbps).")
```

```
            except Exception as e: self.log("Помилка MP3 симуляції: "+str(e)); traceback.print_exc()
        def start_extract(self):
            if self._busy_lock.locked(): messagebox.showinfo("Зайнято","Інша операція виконується."); return
            threading.Thread(target=self._extract_thread,daemon=True).start()
        def _extract_thread(self):
            with self._busy_lock:
                try: self._extract_message()
                except Exception as e: self.log("Помилка extract: "+str(e)); traceback.print_exc()
        def _extract_message(self):
            if self.stego_audio is None: messagebox.showerror("Помилка","Завантажте стего-аудіо."); return
            key_block=None
            if self.plain_key_block is not None: key_block=self.plain_key_block; self.log("Використовується plain ключ-блок.")
            elif self.encrypted_key_block is not None:
                password=self.extract_key_entry.get().strip()
                if not password: messagebox.showerror("Помилка","Введіть пароль для дешифрування."); return
                try:
                    dec=decrypt_bytes(self.encrypted_key_block,password)
                    key_block=json.loads(dec.decode('utf-8')); self.log("Ключ-блок дешифровано.")
                except: messagebox.showerror("Помилка","Не вдалося дешифрувати ключ-блок."); return
            else: messagebox.showerror("Помилка","Немає ключ-блоку."); return
            mdct_frames=self.mdct(self.stego_audio,self.window_size,self.hop_size)
            frame=mdct_frames[key_block["frame_idx"]]
            extracted_bits=[]
            for idx,pos in enumerate(key_block["positions"]):
                coeff=frame[pos]; delta=key_block["deltas"][idx]
                q=QIMQuantizer(delta=delta,alpha=self.qim_alpha)
                extracted_bits.append(q.dequantize(coeff))
            bits_restored=[int(round(sum([int(b) for b in extracted_bits[i:i+self.redundancy]])/self.redundancy)) for i in range(0,len(extracted_bits),self.redundancy)]
            chars=[chr(int(''.join(map(str,bits_restored[i:i+8])),2)) for i in range(0,min(128,len(bits_restored))//8)]
            message=''.join(chars)
            self.extracted_text.delete("1.0",tk.END); self.extracted_text.insert(tk.END,message)
            ber=sum([bits_restored[i]^int(b) for i,b in enumerate(''.join(format(ord(c),'08b') for c in message)[:128])])/128
            self.ber_label.config(text=f"BER: {ber:.4f}")
        def calculate_snr(self):
            if self.original_audio is None or self.stego_audio is None: return 0
            min_len=min(len(self.original_audio),len(self.stego_audio))
            signal=self.original_audio[:min_len]; noise=self.stego_audio[:min_len]-signal
            return 10*np.log10(np.sum(signal**2)/np.sum(noise**2)+1e-12)
        def plot_spectrogram_original(self):
            if self.original_audio is None: return
            self.figure.clf(); ax=self.figure.add_subplot(111)
            S=np.abs(librosa.stft(self.original_audio));
librosa.display.specshow(librosa.amplitude_to_db(S,ref=np.max),ax=ax)
            ax.set_title("Оригінальна спектрограма"); self.canvas.draw()
        def plot_spectrogram_stego(self):
            if self.stego_audio is None: return
            self.figure.clf(); ax=self.figure.add_subplot(111)
            S=np.abs(librosa.stft(self.stego_audio));
librosa.display.specshow(librosa.amplitude_to_db(S,ref=np.max),ax=ax)
            ax.set_title("Стего спектрограма"); self.canvas.draw()
        def show_metrics(self):
            snr=self.calculate_snr()
            self.log(f"SNR: {snr:.2f} dB")
        def save_stego_audio(self):
            if self.stego_audio is None: messagebox.showerror("Помилка","Немає стего-аудіо для збереження."); return
```

```
        path=filedialog.asksaveasfilename(defaultextension=".wav",filetypes=[("WAV files","*.wav")])
        if not path: return
        sf.write(path,self.stego_audio,self.sr)
        self.log(f"Стего-аудіо збережено: {path}")
        key_path=filedialog.asksaveasfilename(defaultextension=".key",filetypes=[("Key files","*.key")])
        if not key_path: return
        if self.plain_key_block is not None:
            with open(key_path,'w') as f: json.dump(self.plain_key_block,f)
        elif self.encrypted_key_block is not None:
            with open(key_path,'wb') as f: f.write(self.encrypted_key_block)
        self.log(f"Збережено ключ-блок: {key_path}")
if __name__=="__main__":
    root=tk.Tk()
    app=AudioSteganographyApp(root)
    root.mainloop()
```

Додаток В. Ілюстративний матеріал

Вінницький національний технічний університет

Підвищення стійкості методу приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування та шифрування ключ-блоку.

Виконав:
студент 2 курсу, групи 2КІТС-24м
Олексюк Є. М.
керівник д.т.н., професор кафедри МБІС
Яремчук Ю. Є.

Вінниця-2024

Вступ

У сучасних інформаційних системах значна частина даних представлена у мультимедійному форматі. Це підвищує вимоги до їхнього захисту, оскільки традиційні криптографічні методи не приховують сам факт передавання конфіденційної інформації. Стеганографія дає можливість вирішити цю проблему, забезпечуючи одночасно конфіденційність, прихованість та збереження структури даних.

Разом з тим, аудіостеганографічні методи стикаються з труднощами при обробці аудіосигналів у форматі MP3. Втратні перетворення, притаманні MP3-компресії, часто призводять до руйнування вбудованої інформації. Саме тому виникає потреба у створенні методів, здатних зберігати приховані дані після стиснення.

Актуальність та мета

Актуальність:

- У світі постійно зростає обсяг мультимедійних даних.
- Більшість класичних стеганографічних методів нестійкі до MP3-компресії.
- Відсутність стійких методів ускладнює використання аудіостеганографії в практичних системах безпеки.

Мета роботи:
Розробити метод приховування даних, який забезпечує збереження прихованого повідомлення після MP3-стиснення та гарантує криптографічний захист службової інформації.

QIM та шифрування

Quantization Index Modulation (QIM):

Один із найефективніших методів стеганографії у частотній області.

Вбудовування даних здійснюється шляхом зміни індексу квантування спектральних коефіцієнтів.

Сумісний з MDCT, що використовується у MP3-кодеку.

Перевага: висока стійкість до втратних перетворень.

Шифрування ключ-блоку:

Використано алгоритм **AES-256 у режимі CBC**.

Забезпечує надійний захист службової інформації.

Запобігає аналітичним атакам та несанкціонованому відновленню параметрів.

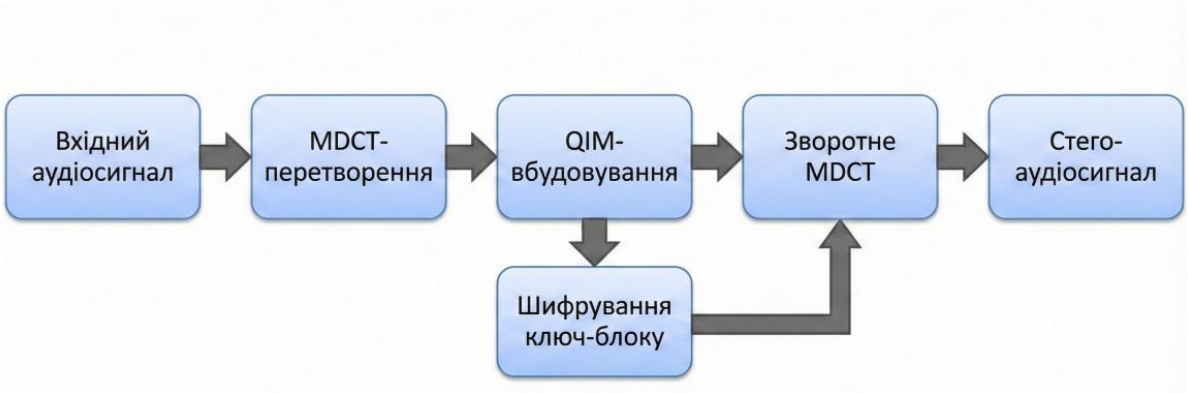
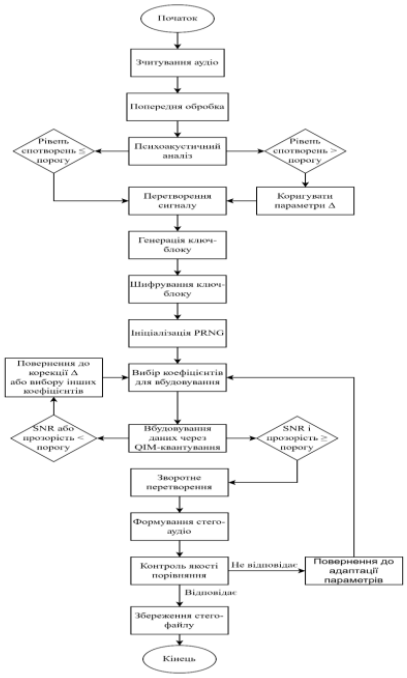


Схема роботи методу



Блок-схема алгоритму

Запропонований гібридний метод

Особливості методу:

Поєднання стеганографії (QIM) та криптографії (AES).

Робота у спектральній області MDCT.

Адаптивне вбудовування відповідно до психоакустичних характеристик.

Збереження непомітності стего-аудіо.

Підвищена стійкість до MP3-обробки.

Система шифрування

У запропонованому методі для захисту службового ключ-блоку використовується симетричний криптоалгоритм **AES-256**. Шифрування виконується в режимі **CBC (Cipher Block Chaining)**, який забезпечує високий рівень стійкості до криптоаналізу.

IV — це ініціалізаційний вектор, випадково згенерована послідовність байтів, яка використовується під час першого раунду шифрування.
Основні властивості IV:

- IV **унікальний для кожної операції шифрування**;
- забезпечує непередбачуваність результату навіть для однакових вхідних даних;
- захищає від атак, що базуються на повторюваних шаблонах повідомлення;
- IV не є секретним, але має бути випадковим і незміненим.

Переваги застосування AES-256 та IV:

Надійний захист ключ-блоку від підміни та несанкціонованого відновлення.

Унеможливлюється відновлення структури прихованого повідомлення без секретного ключа.

Забезпечується **стійкість стеганосистеми при передаванні через незахищені канали**.

Підвищується загальна криптостійкість та стабільність роботи стеганографічного методу.

Тестування

Умови тестування:

Сигнали: музика, мова, змішаний контент.

Перевірка після MP3 з різними бітрейтами:
– 320 кбіт/с
– 192 кбіт/с
– 128 кбіт/с

Результати:

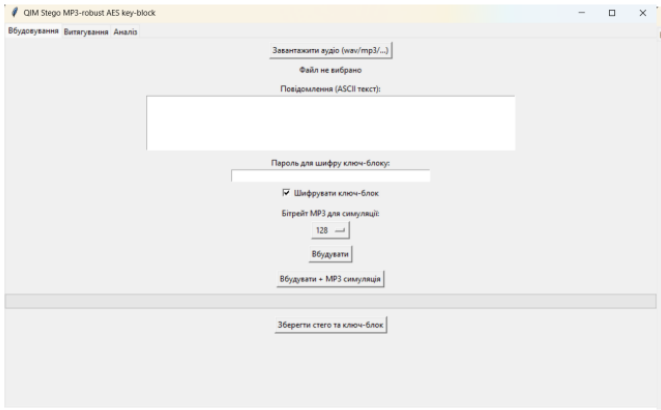
Значне зменшення кількості бітових помилок.

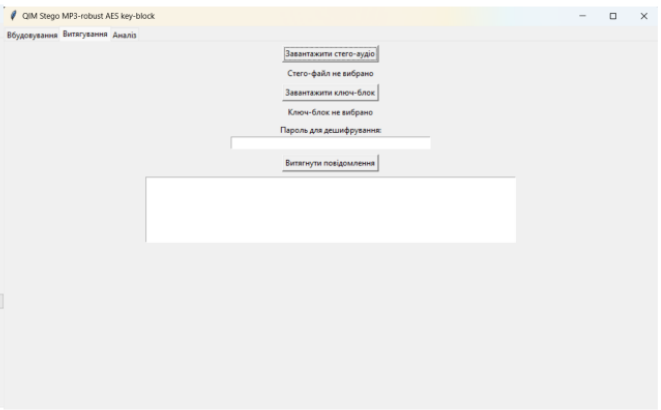
Збереження структури повідомлення після стиснення.

Якість стего-аудіо відповідає початковому сигналу на слух.

Інтерфейс програми

Програму реалізовано на Python.
Інтерфейс побудований таким чином, щоб користувач міг легко виконувати всі основні операції, навіть без досвіду у стеганографії.





У ході виконання роботи було розроблено метод приховування даних в аудіосигналах, спрямований на підвищення стійкості до втратного MP3-стиснення та забезпечення криптографічного захисту службової інформації.

Основні результати дослідження:

- Обґрунтовано доцільність використання QIM-квантування у частотній області MDCT як основи для стеганографічного методу, що сумісний із принципами роботи MP3-кодека.
- Запропоновано гібридний підхід, який поєднує стеганографічні та криптографічні механізми, зокрема шифрування ключ-блоку за алгоритмом AES-256.
- Реалізовано адаптивне вбудовування даних із урахуванням психоакустичних властивостей сигналу, що дозволило зберегти непомітність стего-аудіо та підвищити робастність методу.
- Створено програмний засіб, що забезпечує вбудовування, витягування та аналіз прихованих даних, має зручний графічний інтерфейс і може використовуватися у практичних задачах.
- Результати тестування підтвердили, що запропонований метод зберігає приховані дані після MP3-компресії з різними бітрейтами та демонструє меншу кількість бітових помилок порівняно з класичними методами.

Висновки

Дякую за увагу!

Додаток Г. Протокол перевірки на антиплагіат

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Підвищення стійкості методу приховування даних в аудіосигналах до стиснення MP3 на основі QIM-квантування у частотній області та шифрування ключ-блоку

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра менеджменту та безпеки інформаційних систем
факультет менеджменту та інформаційної безпеки
гр.2КІТС-24м

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism (КП1) **1,18 %**

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

☐ **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак академічного плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту**

☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

☐ У роботі виявлено ознаки академічного плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. Робота до захисту не приймається.

Експертна комісія:

к.т.н., доцент, зав. каф. МБІС Карпінець В.В. _____

к.ф.-м.н., доцент каф. МБІС Шиян А.А. _____

Особа, відповідальна за перевірку Коваль Н.П. _____

З висновком експертної комісії ознайомлений(-на)

Керівник _____ проф. Яремчук Ю.Є.

Здобувач _____ Олексюк Є.М.