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

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Вдосконалення багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації

Виконав: здобувач 2-го курсу,
групи 2КІТС-24м
спеціальності 125– Кібербезпека
та захист інформації
Освітня програма – Кібербезпека
інформаційних технологій та систем
(шифр і назва напряму підготовки, спеціальності)
Волос В.С.
(прізвище та ініціали)
Керівник:
Карпінець В.В.
(прізвище та ініціали)
«09» 12 2025 р.
Опонент:
Тарновський М.Г.
(прізвище та ініціали)
«09» 12 2025 р.

Допущено до захисту
Голова секції УБ кафедри МБІС
Юрій ЯРЕМЧУК
“09” 12 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
"09" 12 2025 р.

З А В Д А Н Н Я
на магістерську кваліфікаційну роботу студенту
Волосу Віталію Сергійовичу
(прізвище, ім`я, по-батькові)

1. Тема роботи Вдосконалення багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації
Керівник роботи Карпінець Василь Васильович
(прізвище, ім`я, по-батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "24" вересня 2025 року № 313
2. Строк подання студентом роботи «_» грудня 2025 року
3. Вихідні дані до роботи: Електронні джерела, наукові статті, книги та документи, повязані з темою. Існуючі теоретичні напрацювання, існуюче програмне забезпечення, технічна документація, вимоги та обмеження до програмного забезпечення
4. Зміст текстової частини: Для досягнення мети з вдосконалення багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації, було поставлено наступні задачі: здійснити аналіз стеганографії, провести огляд сучасних методів реалізації стеганографії, провести аналіз та обгрунтування вдосконалення багатоканального стеганографічного методу, побудувати запропоновані алгоритми роботи системи, обрати мову програмування, та середовище розробки, програмно реалізувати вдосконалений багатоканальний стеганографічний метод з використанням динамічного розподілу прихованої інформації, провести тестування розробленого вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації, здійснити оцінку комерційного потенціалу, здійснити аналіз прогнозування витрат на виконання науково-дослідної роботи, виконати розрахунок економічної ефективності науково-дослідної роботи, здійснити розрахунок ефективності вкладених інвестицій та періоду їх окупності.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов`язкових креслень)

блок-схема загального алгоритму приховування інформації у вдосконаленому багатоканальному стеганографічному методі з динамічним розподілом інформації, блок-схема алгоритму динамічного розподілу інформації у вдосконаленому багатоканальному стеганографічному методі з динамічним розподілом інформації, блок-схема алгоритму шифрування розподіленої інформації у вдосконаленому багатоканальному стеганографічному методі з динамічним розподілом інформації.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина	Карпінець В. В., к.т.н., доц. каф. МБІС		
Економічна частина	Ратушняк О.Г., к.т.н., доцент кафедри ЕПВМ		

7.Дата видачі завдання 24 вересня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи		Примітка
1	Визначення напрямку магістерської кваліфікаційної роботи, формулювання теми	23.05.2025	04.09.2025	Виконано
2	Аналіз предметної області обраної теми	04.09.2025	12.09.2025	Виконано
3	Визначення і обгрунтування вдосконалення багатоканального стеганографічного методу	19.09.2025	26.09.2025	Виконано
4	Розробка запропонованих алгоритмів роботи	26.09.2025	09.10.2025	Виконано
5	Програмна реалізація	10.10.2025	05.11.2025	Виконано
6	Виконання економічної частини	13.11.2025	04.12.2025	Виконано
7	Попередній захист магістерської кваліфікаційної роботи	19.11.2025	19.11.2025	Виконано
8	Захист магістерської кваліфікаційної роботи	08.12.2025	08.12.2025	Виконано

Студент _____ (підпис) Волос В.С. (прізвище та ініціали)

Керівник роботи _____ (підпис) Карпінець В.В. (прізвище та ініціали)

АНОТАЦІЯ

Дана дипломна робота присвячена вдосконаленню багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації. Метою роботи є дослідження та вдосконалення.

Під час розробки та вдосконалення багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації, було розроблено запропоновані алгоритми для покращеної його роботи, та побудовано відповідні блок-схеми .

Програмна реалізація вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації. Метою роботи є дослідження та вдосконалення реалізована в середовищі розробки Visual Studio Code, за допомогою мови програмування JavaScript

В ході роботи, було проведено аналіз існуючих методів компютерної стеганографії, проаналізовано переваги та недоліки кожного з них, також було визначено проблематику теми та сформовано мету роботи, обґрунтовано її цінність.

В результаті виконаної роботи, було проведено різноманітні тестування працездатності, стійкості і захищеності програмно реалізованого вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Також було проведено і розраховано економічну ефективність розробки.

Ключові слова: стеганографія, компютерна стеганографія, багатоканальний метод стеганографії.

ABSTRACT

This diploma thesis is devoted to the improvement of a multichannel steganographic method using dynamic distribution of hidden information. The aim of the work is to research and enhance the method.

During the development and enhancement of the multichannel steganographic method with dynamic distribution of hidden information, proposed algorithms were designed to improve its performance, and the corresponding block diagrams were constructed.

The software implementation of the improved multichannel steganographic method using dynamic distribution of hidden information was developed in the Visual Studio Code environment with the use of the JavaScript programming language.

In the course of the work, an analysis of existing computer steganography methods was carried out, the advantages and disadvantages of each were examined, the main challenges of the topic were identified, and the purpose of the research was formulated and justified.

As a result of the completed work, various tests were conducted to evaluate the functionality, stability, and security of the software-implemented improved multichannel steganographic method using dynamic distribution of hidden information.

Additionally, the economic efficiency of the development was calculated.

Keywords: steganography, computer steganography, multichannel steganographic method.

ЗМІСТ

ВСТУП

Актуальність теми: сучасний розвиток інформаційних технологій, глобалізація цифрових комунікацій та постійне зростання обсягів переданої інформації зумовлюють підвищення вимог до захисту даних. В умовах активного поширення кіберзагроз, спрямованих на перехоплення, модифікацію або підміну інформації, особливого значення набувають методи прихованого передавання даних, зокрема стеганографічні підходи. Дослідження з удосконалення методів стеганографії відповідають пріоритетним напрямам наукових програм України у сфері інформаційної та кібербезпеки, а також входять до переліку тематичних досліджень кафедри, пов’язаних із розробленням захищених систем обміну інформацією та мінімізацією ризиків несанкціонованого доступу.

Мета і задачі дослідження: мета дослідження полягає у вдосконалені багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації, з метою підвищення прихованості, адаптивності та стійкості до атак на виявлення.

Для досягнення поставленої мети визначено наступні задачі:

- проаналізувати існуючі багатоканальні стеганографічні методи та визначити їхні недоліки;
- сформувати концептуальну модель удосконаленого методу з динамічним розподілом інформації;
- розробити алгоритмічне забезпечення функціонування методу;
- провести експериментальні дослідження ефективності запропонованих рішень;
- оцінити стійкість методу до статистичних та структурних атак;

Об’єкт дослідження: об’єктом дослідження є процеси прихованого передавання інформації у цифрових системах з використанням багатоканальних стеганографічних методів.

Предмет дослідження: предметом дослідження є вдосконалені механізми динамічного розподілу інформації між каналами у багатоканальних стеганографічних системах, що забезпечують підвищення прихованості та стійкості до атак на виявлення.

Наукова новизна: наукова новизна отриманих результатів полягає у розробленні вдосконаленого багатоканального стеганографічного методу, у якому вперше застосовано механізм динамічного розподілу інформації між каналами залежно від їх статистичних характеристик та рівня стеганографічного навантаження, також такий метод використовує різнотипні канали. Це дозволило підвищити стійкість до виявлення, збільшити пропускну здатність та адаптивність методу за умов зміни характеристик носіїв.

Практична цінність: практичне значення результатів роботи полягає у можливості використання розробленого методу в системах захищеного документообігу, прихованих каналах зв'язку, інтелектуальних системах кіберзахисту та для підвищення прихованості обміну інформацією в мережах загального користування. Запропоноване програмне забезпечення може бути інтегроване у корпоративні системи захисту інформації.

Особистий внесок магістранта: особистий внесок магістранта полягає в самостійному проведенні аналітичного огляду, розробці моделі та алгоритмічного забезпечення методу, програмній реалізації, постановці та виконанні експериментів, аналізі результатів і формуванні висновків.

Апробація: тези доповідей представлені на Міжнародній науково-практичній Інтернет-конференції студентів, аспірантів та молодих науковців «Молодь у науці: дослідження, проблеми, перспективи (МН-2026)» [1].

1. АНАЛІЗ СТЕГАНОГРАФІЇ ТА ОГЛЯД СУЧАСНИХ МЕТОДІВ ЇЇ РЕАЛІЗАЦІЇ

В даному розділі буде проаналізовано основні засади поняття стеганографії та її роль у забезпечення інформаційної безпеки. Буде проведено огляд сучасних методів стеганографії, окреслено переваги та недоліки кожного з них. Визначено перспективи розвитку та вдосконалення вже існуючих рішень. Та поставлено задачу для виконання в наступних розділах даної магістерської роботи.

1.1. Основи стеганографії та її роль у забезпеченні інформаційної безпеки.

Для того щоб зрозуміти поняття стеганографії повною мірою спершу необхідно визначити поняття криптографії. Після чого поняття стеганографії, її завдання і роль в забезпеченні інформаційної безпеки. Визначити і провести аналіз вже розроблених методів, здійснити їх аналіз з визначеними перевагами та недоліками буде зрозуміло краще [2].

Криптографія – наука про математичні методи забезпечення конфіденційності, цілісності і автентичності інформації. Розвинулась з практичної потреби передавати важливі відомості найнадійнішим чином. Для математичного аналізу криптографія використовує інструментарій абстрактної алгебри та теорії ймовірностей [3].

Криптографічний алгоритм, або шифр, математична формула, що описує процеси шифрування і розшифрування. Щоб зашифрувати відкритий текст, криптоалгоритм працює в сполученні з ключем словом, числом або фразою. Те саме повідомлення одним алгоритмом, але різними ключами буде перетворюватися в різний шифротекст. Захищеність шифротексту цілком залежить від двох речей: стійкості криптоалгоритму і таємності ключа. Криптоалгоритм плюс усілякі ключі і протоколи, що приводять їх у дію, складають криптосистему. PGP – криптосистема [4].

В кінцевому вигляді зашифроване повідомлення виглядає як шифр (спотворене повідомлення), яке не може бути дешифрованим зловмисником якщо він не знає ключа (коду) для його дешифрування [5].

Стеганографія ж відрізняється тим що приховується сам факт передачі прихованої інформації, тобто зловмисник навіть не зрозуміє і не помітить що інформація була прихована і передана [6].

Перші сліди стеганографічних методів губляться в глибокій давнині. Наприклад, відомий такий спосіб приховування письмового повідомлення: голову раба голили, на шкірі голови писали повідомлення і після відростання волосся раба відправляли до адресата [7].

З детективів відомий сучасний метод «мікроточки»: повідомлення записується за допомогою сучасної техніки на дуже маленький носій – «мікроточку», яка пересилається зі звичайним листом, наприклад, під маркою або десь в іншому заздалегідь обумовленому місці.

Один типовий стеганографічний прийом тайнопису – акровірш – добре відомий знавцям поезії. Акростих – це така організація віршованого тексту, при якій, наприклад, початкові літери кожного рядка утворюють приховане повідомлення [8].

В даний час у зв'язку з широким поширенням комп'ютерів відомо багато тонких методів «захована» інформації, що захищається, всередині великих обсягів інформації, що зберігається в комп'ютері [9].

Навіть з наведеної невеликої кількості прикладів видно, що при використанні стеганографії, на відміну від криптографії, інформація, що захищається, не перетворюється, а приховується сам факт її передачі.

В кінці 90 років стеганографію поділили на 3 види: класичну, комп’ютерну і цифрову, також виділяють мережеву. Класична пов`язана із приховуванням текстових даних, використовуючи властивості самого тексту або ж навколишнього середовища. З іншого боку – комп’ютерна і цифрова стеганографія, які розглядають методи приховування будь-якої

електронної інформації (текст, звуковий файл, зображення, відео, програма) з використанням можливостей інформаційних систем. Мережева стеганографія використовує можливості протоколу передачі даних транспортного рівню – TCP [10].

Класична стеганографія. В часи другої світової війни використовували такий тип стенографії як мікроточки (мікроскопічні фотографії наклеєні в текст посилання). Також прикладом класичної стеганографії можна назвати надписи на бокових колодах карт та будьякий тип жаргонного шифру, де слова мають обґоворені значення, семаграми [11].

Одним з найпоширеніших методів класичної стеганографії є використання симпатичних чорнил (невидимих). Зазвичай процес запису здійснюється наступним чином: перший шар – наноситься важливий запис невидимим чорнилом, другий шар – запис видимими чорнилом, що нічого не значить. Текст, записаний такими чорнилом, проявляється лише за певних умов (нагрівання, освітлення, хімічний проявник і т.ін.). Існує також чорнило з хімічно нестабільним пігментом. Написане цими чорнилами виглядає як написане звичайною ручкою, але через певний час нестабільний пігмент розкладається, і від тексту не залишається і сліду [12].

Комп'ютерна стеганографія – напрям класичної стеганографії, заснований на особливостях комп'ютерної платформи. Комп'ютерна стеганографія базується на двох основних принципах. Перший принцип полягає в тому, що файли, що містять оцифроване зображення або звук, можуть бути до певної міри видозмінені без втрати їх функціональності на відміну від інших типів даних, що вимагають абсолютної точності. Другий принцип полягає в нездатності органів почуттів людини розрізняти незначні зміни в кольорі зображення або якості звуку. Цей принцип особливо легко застосовувати до зображення або звуку, який несе надлишкову інформацію [13].

Цифрова стеганографія – напрям класичної стеганографії, заснований на захованні або впровадженні додаткової інформації в цифрові об'єкти, викликаючи при цьому деякі спотворення цих об'єктів. Але, як правило, дані об'єкти є мультимедіа-об'єктами (зображення, відео, аудіо, текстури 3D-об'єктів) та внесення спотворень, які знаходяться нижче межі чутливості середньостатистичної людини, не призводить до помітних змін цих об'єктів [14].

Комп’ютерна стеганографія здійснюється різними способами. Загальною ж рисою таких способів є те, що приховуване повідомлення вбудовується в об’єкт, що не привертає увагу і потім відкрито транспортується (пересилається) адресатові [15].

У сучасній комп’ютерній стеганографії існує два основних типи файлів: повідомлення - файл, що призначений для приховування, і контейнер – файл, що може бути використаний для приховування в ньому повідомлення. При цьому контейнери бувають двох типів. Контейнер-оригінал (або "порожній" контейнер) – це контейнер, що не містить схованої інформації. Контейнер-результат (або "заповнений" контейнер) – це контейнер, що містить сховану інформацію. Під ключем розуміється секретний елемент, що визначає порядок занесення повідомлення в контейнер. Класичним є наступний принцип вбудовування даних [16].

Нехай сигнал контейнера представлений послідовністю з N біт. Процес приховання інформації починається з визначення біт контейнера, які можна змінювати без внесення помітних спотворень – стеганошляху. Далі серед цих біт, зазвичай у відповідності до ключа, обираються біти, що замінюються бітами приховуваного повідомлення (рис. 1.1) [17].

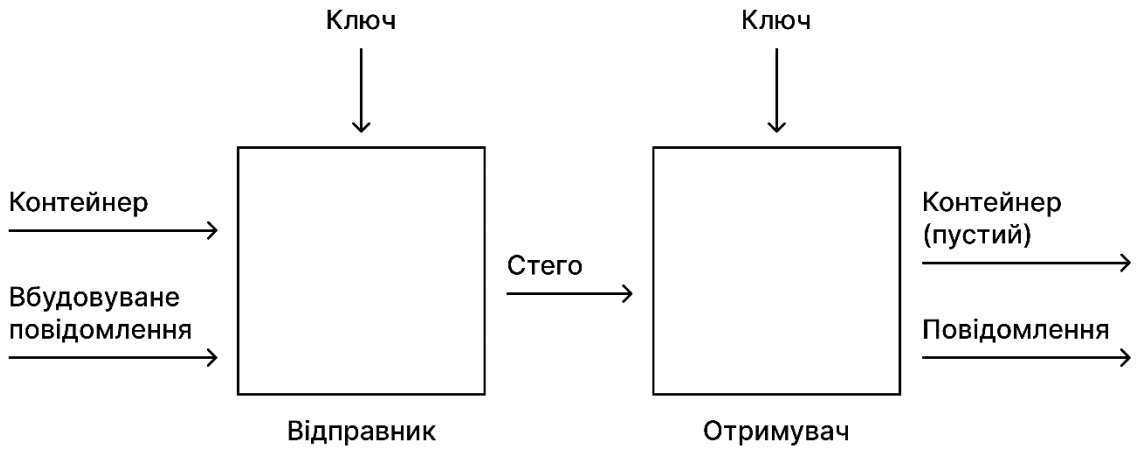


Рисунок 1.1 – процес приховування інформації стеганографічним методом.

Базовими положеннями сучасної комп’ютерної стеганографії є [18]:

1. Забезпечення автентичність і цілісність файлу.

2. Обізнаність супротивника з можливостями комп’ютерної стеганографії. 3. Безпека ґрунтується на збереженні стеганографічним перетворенням основних властивостей переданого файлу при внесенні в нього секретного повідомлення і деякої невідомої супротивникові інформації - ключа.

4. Витяг секретного повідомлення має становити складну обчислювальну задачу.

1.2. Аналіз існуючих методів комп’ютерної стеганографії реалізації та окреслення переваг та недоліків кожного з них.

Стеганографічні методи, що застосовуються для приховування інформації, різняться за принципом дії, та характеризуються трьома якісними параметрами, а саме непомітність, ємність і стійкість [19].

Непомітність є найважливішим аспектом оцінки якості методу приховування інформації. Непомітність оцінюється як різниця між оригінальним контейнером та стегоконтейнером. Існує низка показників

для оцінки відмінностей стегоконтейнера від оригінального контейнера. До таких показників відносять сенсорні і статистичні дані (тобто ті зміни які здатне побачити людське око) [20].

Ємність показує, скільки даних може бути приховано в носії. Ця характеристика залежить від стеганографічного алгоритму та властивостей контейнера. Існують різні метрики для вимірювання місткості, для прикладу кількість бітів секретного повідомлення, які можна вбудувати в один піксель зображення або відношення розміру секретного повідомлення до максимального розміру повідомлення, що може бути вбудоване в конкретний контейнер [21].

Стійкість означає невразливість до модифікацій стегоконтейнера, наприклад, стиснення, перетворення в інший формат або часткового пошкодження тощо. Це здатність зберігати приховану інформацію у первісному вигляді навіть після різних видів обробки. До прикладу таких як фільтрація, додавання шуму, зміна розміру, повороти, стиснення, або інші геометричні та нелінійні перетворення. Стійкість також включає здатність протистояти спробам виявлення та витягнення прихованих даних за допомогою статистичних або криптографічних методів [22].

Усі ці параметри можуть бути конфліктними, їх взаємовплив та взаємозв’язок залежить від конкретних застосувань, і потреб при виборі самого методу. Якщо потрібна велика ємність то більша частина носія використовується для секретного зберігання даних. Це дозволяє передавати більше даних, але водночас призводить до більших спотворень файлу-носія. Таким чином, оптимальний вибір параметрів залежить від конкретних вимог і пріоритетів стеганографічного застосування (рис. 1.2) [23].

Рисунок 1.2 – основні принципи методів стеганографії.

З розвитком цифрових технологій та засобів обробки даних виникла потреба у вдосконаленні класичних підходів і створенні нових, більш надійних методів стеганографії.

Серед найпопулярніших методів стеганографії, виділяють наступні: метод найменш значущого біту (LSB – Least Significant Bit); метод дискретного косинусного перетворення (DCT – Discrete Cosine Transform); метод дискретного вейвлет-перетворення (DWT – Discrete Wavelet Transform) та метод PVD (Pixel Value Differencing).

Огляд стеганографічного методу найменш значущих бітів (Least Significant Bit LSB):

LSB-стеганографія – це базовий метод, який приховує інформацію в зображеннях, відео чи аудіофайлах, замінюючи найменш значущий біт кожного пікселя чи зразка бітом із секретного повідомлення. Однак він може бути незахищеним від передових методів виявлення чи атак. Інформація, прихована в зображенні, має на меті змінити найменш значущі біти в зображенні-носії на потрібну інформацію. LSB є найбільш використовуваним у стеганографії. Це один із найпростіших та широко використовуваних методів стеганографії. Ідея цього методу полягає в тому, що найменші компоненти зображення надають лише дуже малу частку інформації яку людське око не може навіть помітити (рис. 1.3) [24].

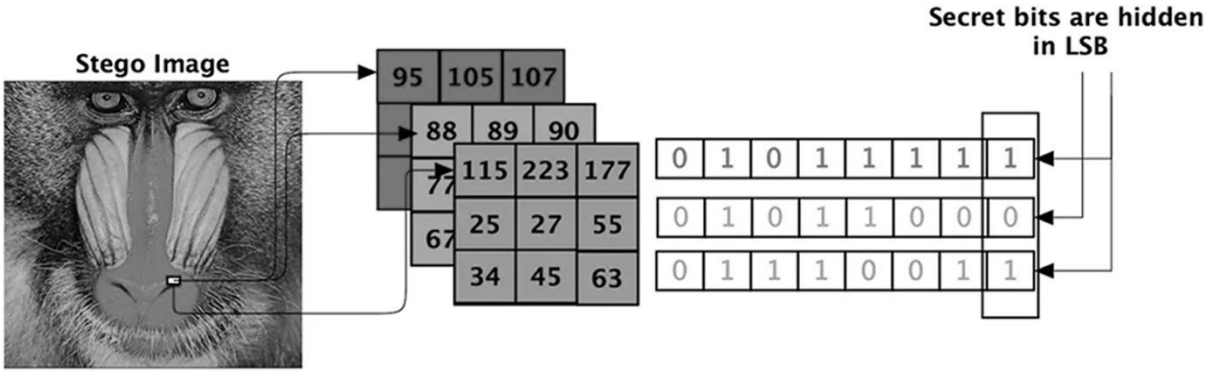


Рисунок 1.3 – приклад стеганографічного методу комп’ютерної стеганографії найменш значущих бітів (Least Significant Bit LSB).

У більшості запропонованих методів біти секретного повідомлення випадковим чином або послідовно вставляються в молодший біт (LSB) позицій пікселів.

Переваги стеганографічного методу комп’ютерної стеганографії найменш значущих бітів (Least Significant Bit LSB) [25]:

- це найпростіший стеганографічний метод вбудовування повідомлення в цифровий носій, такий як зображення або звук;
- він не вимагає перетворення середовища обкладинки;
- заміна LSB гарантує, що стего-носій не відрізняється від оригінального середовища обкладинки;
- це найшвидший метод за швидкістю вбудовування;

І навпаки, основними недоліками LSB-стеганографії є:

- стего-носій вразливий до відносно простого статистичного аналізу;
- приховані дані легко знищуються шляхом простої маніпуляції або обробки стего-об'єкта;
- легко вставляти дані, порівнюючи величину значення пікселя або абсолютне значення LSB із секретними даними;

Огляд стеганографічного методу дискретного косинусного перетворення (Discrete Cosine Transform DCT):

В алгоритмі Discrete Cosine Transform відбувається стиснення JPEG, тому методи стеганографії на основі DCT застосовуються лише для формату зображення jpeg. Принцип дії методу комп'ютерної стеганографії на основі дискретного косинусного перетворення DCT базується на тому що зображення-носій формату JPEG послідовно розділяється на блоки розміром 8×8 пікселів. Після чого кожен блок перетворюється з просторової області (де значення пікселів представляють яскравість або інтенсивність кольору) у частотну область за допомогою дискретного косинусного перетворення (DCT). Це перетворення розділяє інформацію зображення на частотні коефіцієнти, які описують, скільки кожної частоти є в цьому блоці, причому нижчі частоти представляють більш плавні зміни, а вищі частоти, швидкі зміни яскравості або деталізації зображення [26].

На наступному етапі у частотній області, алгоритм обирає певні коефіцієнти в які вбудовуються секретні дані. Зазвичай модифікуються менш значущі або середньочастотні коефіцієнти, оскільки зміни в цих частотах менш помітні для людського ока. В даному алгоритмі як правило уникають найнижчих частотних коефіцієнтів (які впливають на загальний вигляд зображення) і найвищих частотних коефіцієнтів (які можуть бути втрачені під час стиснення). Секретні біти вбудовуються шляхом незначного змінення значень цих вибраних коефіцієнтів, наприклад, шляхом зміни їх найменш значущих бітів або коригування їх величини відповідно до секретних даних [27].

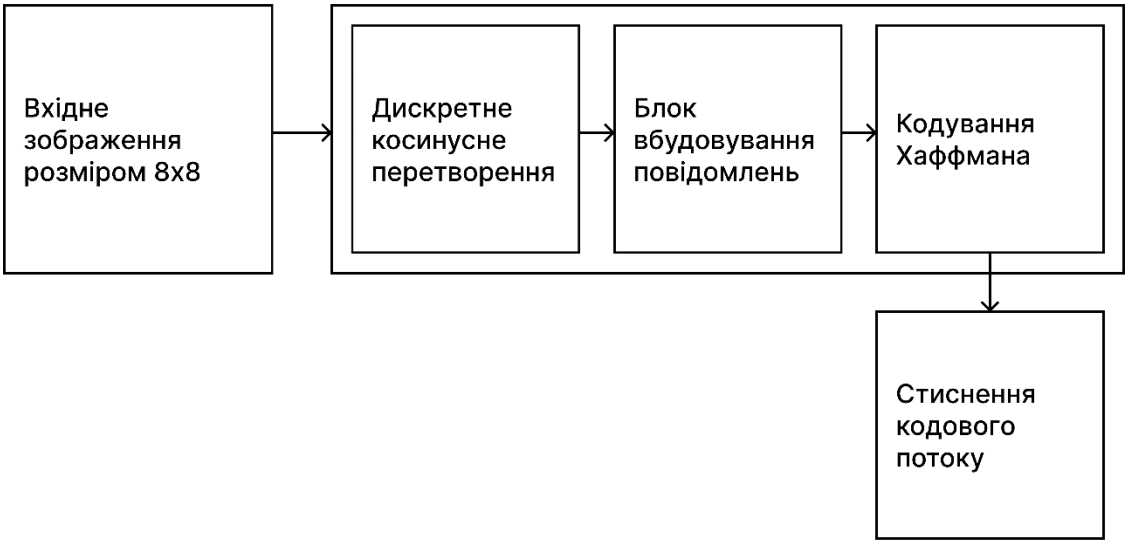


Рисунок 1.4 – алгоритм роботи стеганографічного методу дискретного косинусного перетворення (Discrete Cosine Transform DCT).

Після вбудовування модифіковані коефіцієнти перетворюються в зворотньому порядку назад у просторову область за допомогою оберненого дискретного косинусного перетворення (IDCT). У результаті отримуємо кінцеве стего-зображення, яке візуально ідентичне або майже ідентичне оригінальному зображенню (носію), але тепер містить приховану інформацію у своїй частотній структурі [28].

Огляд стеганографічного методу мультифрагментної стеганографії (Multi-image steganography):

Multi-image steganography – це схема приховування даних, за допомогою якої користувач намагається приховати конфіденційні повідомлення в декількох зображеннях. На відміну від традиційної стеганографії, яка вимагає лише безпеки окремого зображення, Multi-image steganography враховує загальну безпеку групи зображень [29].

Однак існуючі Multi-image steganography методи стикаються з нетривіальною проблемою, так як інформація розділяється порівно між носіями не враховуючи їх особливості. Саме тому постає проблема, як оптимально та динамічно розподілити корисне навантаження між

декількома носіями, щоб підвищити безпеку приховуваної інформації (рис. 1.5) [30].

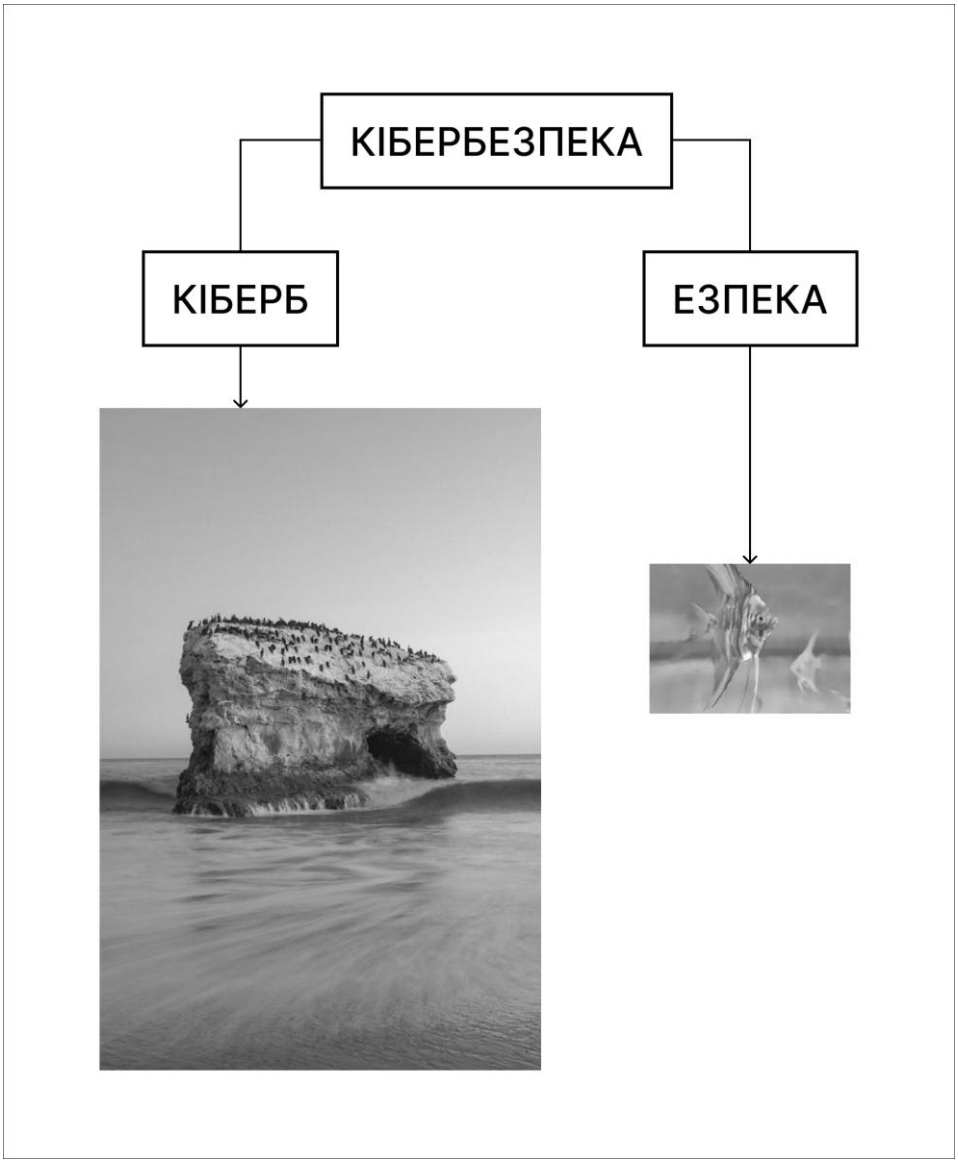


Рисунок 1.5 – приклад розподілу приховуваної інформації між 2-ма каналами в методі мультифрагментної стеганографії Multi-image steganography.

Вирішення цієї проблеми буде запропоновано в наступних розділах даної робот, а саме розробка багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Здійснемо порівняльну характеристику описаних методів комп’ютерної стеганографії (таблиця. 1).

Таблиця 1 – порівняльна характеристика описаних методів комп’ютерної стеганографії [31].

Критерій оцінки	Метод комп’ютерної стеганографії		
	Least Significant Bit (LSB)	**Discrete Cosine Transform (DCT)**	**Multi-image steganography**
Принцип роботи	Приховування інформації в найменш значущих бітах пікселів файлів.	Дані вставляються у коефіцієнти DCT (частотна область) після JPEG-перетворення.	Повідомлення ділиться на рівні частини, кожна з яких ховається у різних зображеннях.
Тип середовища	Універсальний -може використовуватися в растрових зображеннях, аудіо або навіть відео.	Лице зображення формату JPEG	Потребує кількох носіїв (кількох зображень).
Формат носія	BMP, PNG, TIFF, JPEG, MPEG, MP4 тощо.		
Видимість змін	Дуже низька, зміни невидимі для людського ока при коректному використанні.	Майже непомітна, але при високій інтенсивності змін (великій кількості приховуваних даних) може впливати на якість.	Залежить від кількості носіїв і розмірі приховуваної інформації

Ємність (обсяг даних, які можна сховати)	Дуже висока, дозволяє зберігати великі обсяги даних без помітних змін.	Середня, обмежена кількістю коефіцієнтів, придатних до зміни.	Залежить від кількості зображень-носіїв які використовуються.
Основна перевага	Висока ємність, універсальність.	Стійкість до стиснення	При великій кількості носіїв можна помістити дуже багато інформації
Основний недолік	В порівнянні з іншими методами, менша стійкість до стиснення.	Менша ємність, та типи файлів носіїв (лише зображення формату JPEG)	Використовує рівномірний розподіл даних між носіями.

В порівняльному аналізі описаних вище методів і моделей комп’ютерної стеганографії, можна дійти висновку що в кожного методу є свої суттєві переваги та недоліки, при виборі кожного з них важливо опиратися на задачі які він повинен виконати та за потреби тип файлу-носія.

1.3. Постановка задачі

В результаті даного розділу було визначено проблему у недостатній ефективності існуючих багатоканальних стеганографічних методів, як от (Multi-image steganography) зумовленій відсутністю механізмів динамічного розподілу прихованої інформації між каналами, що значно знижує захищеність інформації і стійкість методу загалом. Проведений аналіз показав, що більшість сучасних методів забезпечують високу ємність або непомітність, проте не дозволяють динамічно і адаптивно змінювати структуру вбудовування залежно від характеристик каналів передавання та їх типів.

Виходячи з проведеного аналізу, можна сформулювати тему роботи: «Вдосконалення багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації».

Для досягнення визначеної мети необхідно виконати наступні завдання:

- побудувати запропоновані алгоритми роботи вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації;
- обрати мову програмування, та середовище розробки для програмної реалізації вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації;
- програмно реалізувати вдосконалений багатоканальний стеганографічний метод з використанням динамічного розподілу прихованої інформації;
- провести тестування розробленого вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації;

1.4. Висновки до розділу 1.

В даному розділі було проаналізовано основні засади поняття стеганографії та її роль у забезпечення інформаційної безпеки. Проведено огляд сучасних методів стеганографії, окреслено переваги та недоліки кожного з них. Визначено проблематику, шляхи і методи покращення вже існуючих рішень. Також в даному розділі було поставлено задачу для виконання в наступних розділах да досягнення остаточної мети даної магістерської роботи, а саме вдосконалення багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

2. ПРОЕКТУВАННЯ ВДОСКОНАЛЕНОГО БАГАТОКАНАЛЬНОГО СТЕГАНОГРАФІЧНОГО МЕТОДУ З ВИКОРИСТАННЯМ ДИНАМІЧНОГО РОЗПОДІЛУ ПРИХОВАНОЇ ІНФОРМАЦІЇ

В даному розділі буде розроблено запропоновані алгоритми роботи вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації, та побудовано відповідні блок-схеми роботи цих алгоритмів, а саме:

- загальний алгоритм роботи вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації;
- алгоритм приховування інформації в зображення;
- алгоритм приховування інформації в аудіо файл;
- алгоритм динамічного розподілу приховуваної інформації.

Також буде спроектовано користувацький інтерфейс для подальшої програмної реалізації вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

2.1. Обґрунтування та аналіз вдосконалення багатоканального стеганографічного методу.

Методи сучасної комп’ютерної стеганографії постійно вдосконалюються та розвиваються, з метою підвищення рівня захисту інформації, збільшення стійкості до атак і збереження непомітності прихованих даних. Одним із таких напрямів розвитку є багатоканальні методи, в яких передбачається розподіл прихованої інформації між кількома носіями (зображеннями, аудіо- або відеофайлами). Проте більшість існуючих підходів, зокрема класичний метод Multi-image Steganography, здійснюють розподіл даних рівномірно, не враховуючи індивідуальні властивості кожного носія (каналу), такі як ємність, формат,

частотний спектр або рівень шуму. Це призводить до неефективного використання ресурсів носіїв і знижує загальний рівень непомітності та стійкості методу [32].

Вдосконалення яке запропоноване в даній магістерській кваліфікаційній роботі, полягає у впровадженні динамічного розподілу приховуваної інформації між каналами під час приховування на основі їх властивостей. Тобто обсяг даних, який приховується у кожному каналі, визначається не рівномірно, а пропорційно до максимальної кількості бітів, які канал може вмістити без втрати якості та без порушення критерію непомітності. Запропонований підхід забезпечить адаптивність системи до різних типів носіїв (зокрема фото та аудіо файлів) і підвищить ефективність використання стеганографічного простору шляхом збільшення кількості інформації яка поміститься в обрані носії (канали).

Теоретично процес розподілу повідомлення можна описати через коефіцієнт ємності каналу [33]. Нехай існує множина каналів $C = \{C_1, C_2, \dots, C_n\}$, де кожен канал (носій) характеризується своєю максимальною ємністю Q_i – кількістю бітів, які можуть бути змінені без візуальних або акустичних спотворень (тобто скільки інформації може вмістити кожен з каналів). Тоді загальна ємність системи визначається за формулою 2.1:

$$Q_{\text{заг}} = \sum_{i=1}^{n} Q_i \tag{2.1}$$

Для забезпечення оптимального розподілу приховуваної інформації M обсяг даних, що приховується у кожному каналі, обчислюється пропорційно у співвідношенні до ємності конкретного каналу та загальної ємності носіїв, та розраховується за формулою 2.2:

$$M_i = M * \frac{Q_i}{Q_{\text{заг}}}$$

(2.2)

де M_i обсяг даних, що приховується у i каналі. Дана формула забезпечує динамічний, пропорційний до можливостей кожного каналу розподіл повідомлення, який дозволяє уникнути перевантаження одного каналу і водночас максимально використати потенціал кожного з них.

З теоретичної точки зору, динамічний розподіл також позитивно впливає на показник непомітності та стійкості запропонованого методу в порівнянні з аналогом. Якщо позначити спотворення контейнера як D_i а середнє допустиме спотворення – як D_{max}, тоді умова збереження непомітності набуває вигляду, та розраховується за формулою 2.3:

$$\frac{1}{n}\sum_{i=1}^{n} D_i \leq D_{max}$$

(2.3)

При рівномірному розподілі повідомлення дана умова часто порушується через перевантаження окремих каналів (носіїв). У запропонованому вдосконаленому стеганографічному методі, завдяки динамічному розподілу обсягу обраних даних між каналами значення D_i автоматично корегується залежно від особливостей і властивостей конкретно обраного носія, що дозволяє мінімізувати загальне спотворення стегоконтейнера.

Крім того, застосування різних типів каналів, наприклад зображення та аудіо, сприяє підвищенню стійкості до стеганалізу, оскільки зловмисник має справу не з одним носієм і навіть не з одним типом носіїв, а з різними за форматом носіями даних, де кожен канал має власні характеристики шуму, частотного діапазону та статистичних даних та відхилень. Це ускладнює процес виявлення прихованої інформації в стегоконтейнері, особливо при використанні криптографічного шифрування перед безпосереднім вбудовуванням [34].

Таким чином, основна ідея вдосконалення полягає у тому, що замість фіксованого або рівномірного розподілу інформації між обраними каналами, використовується динамічна модель розподілу даних, що враховує параметри кожного з носіїв. Це дозволяє забезпечити баланс між трьома ключовими характеристиками стеганографії – ємністю, непомітністю та стійкістю, що математично можна представити у вигляді оптимізаційної задачі, формула 2.4:

$$\max F = \alpha * C + \beta * R - \gamma * D \tag{2.4}$$

де C – показник ємності;

R – стійкість;

D – рівень спотворення;

α, β, γ – вагові коефіцієнти, які визначають пріоритетність кожного параметра.

Завдання запропонованого вдосконаленого стеганографічного методу полягає у знаходженні такого динамічного розподілу M_i , який максимізує значення функції F.

Загалом вдосконалений багатоканальний стеганографічний метод із використанням динамічного розподілу прихованої інформації забезпечує більш ефективне використання ємності носіїв (каналів), підвищує стійкість до атак, стеганоаналізу та зменшує помітність змін у контейнерах. На відміну від класичних методів, він дозволяє здійснювати адаптивне, інтелектуальне управління процесом приховування, що значно покращує загальні показники безпеки та якості прихованої передачі інформації.

Для підтвердження доцільності запропонованого вдосконалення, проведено теоретичний аналіз, який порівнює базовий багатоканальний метод Multi-image Steganography з запропонованим вдосконаленнням, що використовує динамічний розподіл прихованої інформації. Порівняння

здійснювалося за ключовими критеріями якості стеганографічних систем: непомітність, ємність, стійкість, гнучкість, а також рівень безпеки.

Класичний метод Multi-image Steganography передбачає рівномірний розподіл повідомлення між кількома носіями, незалежно від їхньої структури, типу або якості. Такий підхід має низку переваг — простоту реалізації, однакове навантаження між каналами та відносно швидку обробку. Проте він не враховує неоднорідність каналів і не забезпечує адаптації системи до зміни їхніх властивостей. У результаті спостерігається надлишкове навантаження на окремі носії, що підвищує ймовірність виявлення прихованої інформації та зменшує загальну стійкість системи.

У вдосконаленому методі запропоновано принципово інший підхід — динамічний розподіл даних, який здійснюється відповідно до характеристик обраних носіїв (каналів). Якщо для i каналу визначено ємність Q_i, то частка інформації, яка в нього вбудовується, обчислюється за формулою 2.5:

$$M_i = M * \frac{Q_i}{Q_{\text{заг}}} \tag{2.5}$$

Це забезпечує адаптивне використання ресурсів кожного носія, мінімізує спотворення контейнерів після приховування інформації та підвищує рівень непомітності, особливо коли використовуються носії різних типів (наприклад, зображення та аудіо). Таким чином, вдосконалений метод є більш захищеним, стеганостійким, універсальним і гнучким рішенням, придатним для складних стеганографічних сценаріїв [35].

Таблиця 2.1 – Порівняння базового методу багатоканальної комп’ютерної стеганографії та вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Критерій оцінки	**Існуючий метод багатоканальної комп'ютерної стеганографії Multi-image Steganography**	**Вдосконалений багатоканальний стеганографічний метод з використанням динамічного розподілу інформації**
Тип розподілу	Рівномірний, без урахування параметрів обраних носіїв	Динамічний, пропорційний до ємності обраних носіїв (каналів)
Типи каналів	Лише однотипні (зображення)	Різнотипні (зображення, аудіо, відео)
Ємність системи	Обмежена, часто не використана, або переповнена ємність одного із каналів	Збільшена та максимально ефективна, використовується кожен канал
Непомітність (PSNR)	Середня, з ризиком перевищення допустимого спотворення	Вища завдяки адаптивному навантаженню на кожен із каналів
Стійкість до атак	Середня, залежить від кількості каналів	Підвищена за рахунок розподілу за типом і шифрування
Складність реалізації	Низька	Середня, потребує обчислення параметрів носіїв, та додаткового шифрування даних перед приховуванням для кращого захисту даних
Безпека	Базова (тільки розподіл)	Підвищена за рахунок шифрування та різнотипних каналів різних форматів і типів.

З наведеного порівняння можна зробити висновок, що вдосконалений багатоканальний стеганографічний метод з використанням динамічного розподілу інформації має перевагу за всіма ключовими параметрами, хоча є дещо складніший в реалізації. Однак збільшення складності компенсується зростанням ефективності, підвищенню безпеки та універсальності запропонованого методу. Особливо важливою є можливість комбінувати різні типи носіїв, що забезпечує суттєве підвищення рівня захисту прихованої інформації в вихідному стегоконтейнері.

Теоретичні результати аналізу показують, що при використанні динамічного розподілу середній показник непомітності може зрости на 10–15%, тоді як коефіцієнт втрати даних при стисненні зменшується приблизно на 8–12%. Це свідчить про те що вдосконалений метод здатен більш ефективно зберігати якість носіїв, забезпечуючи при цьому високу стійкість до різноманітних впливів і статистичного стеганоаналізу.

У результаті проведеного теоретичного порівняння можна зробити висновок, що вдосконалений багатоканальний стеганографічний метод з динамічним розподілом прихованої інформації забезпечує оптимальне поєднання ємності, непомітності та стійкості, перевершуючи показники класичних аналогів. Це робить його перспективним напрямом розвитку сучасних систем приховування інформації у середовищі з підвищеними вимогами до безпеки даних.

2.2 Вибір та обґрунтування стеганографічних каналів у вдосконаленому багатоканальному стеганографічному методі з використанням динамічного розподілу прихованої інформації.

У процесі розроблення вдосконаленого багатоканального стеганографічного методу одним із ключових завдань стало визначення оптимальних каналів для приховування інформації. Вибір каналів

безпосередньо впливає на стійкість системи, приховану ємність та рівень непомітності внесених змін. Оскільки метод передбачає динамічний розподіл прихованої інформації, важливо обирати такі типи контейнерів, які забезпечують можливість адаптивного перерозподілу навантаження і при цьому залишаються малопомітними для візуального чи акустичного сприйняття. Тому сформовано порівняльну таблицю найпопулярніших каналів комп'ютерної стеганографії Таблиця 2.2 [36, 37, 38].

Таблиця 2.2. Порівняльна таблиця найпопулярніших каналів комп'ютерної стеганографії.

Критерії	Зображення	Аудіо	Відео	Текст
Прихована ємність	Висока, великі обсяги даних завдяки кількості пікселів	Середня, залежить від частоти дискретизації	Дуже висока, багато кадрів + звук	Низька, дуже мало місця
Рівень непомітності	Дуже висока, зміни в молодших бітах непомітні для ока	Висока, слух не помічає низькорівневих змін	Дуже висока, приховування розподіляється у просторі та часі	Низька, оскільки зміни можуть бути помітні
Стійкість до атак	Середня, вразливість до статистичних атак	Висока, важче аналізувати акустичні сигнали	Висока, важко аналізувати весь потік	Середня
Вразливість до перетворень	Висока, змінюється при JPEG-компресії	Середня, вразливе до перекодування MP3	Низька, стиснення відео сильно руйнує дані	Дуже висока, будь-яке форматування руйнує дані
Переваги	Простота реалізації,	Важко виявити, хороша стійкість	Максимальна ємність,	Легкість передачі,

	велика ємність		складність для аналізу	невеликі розміри
Недоліки	Нестійке до втрат при стисненні та фільтрації	Може спотворюватися після обробки аудіоредакторами	Складність реалізації та великий розмір файлів	Найменша ємність, найвища помітність

З огляду на це було обрано два основних канали такі як цифрове зображення та аудіофайл, які найбільше підходять під потреби окресленого вище завдання.

Зображення традиційно вважається одним із найбільш ефективних каналів для стеганографії через значну кількість пікселів, що дозволяє приховувати великі обсяги інформації за рахунок мінімального внесення змін у низькорівневі бітові структури. У межах запропонованого методу зображення використовується як один із основних носіїв, оскільки модифікації окремих пікселів є практично непомітними для людського ока. Візуальна система людини не здатна розрізнити незначні зміни у молодших бітах компонентів кольору, що створює можливість приховувати інформацію у просторі RGB без ризику появи видимих артефактів. Додатковою перевагою є можливість адаптивно змінювати інтенсивність приховування в різних частинах зображення: текстурні та насичені області дозволяють вбудовувати більше даних, тоді як однорідні фрагменти з низькою варіативністю потребують обережного втручання. Такий підхід забезпечує природну адаптацію методу до локальних характеристик контейнера та підвищує загальний рівень непомітності.

Другим каналом приховування інформації є аудіофайл. Аудіостеганографія має свої унікальні властивості, які роблять її цінним доповненням до візуальних каналів. Аудіосигнал змінюється у часі, тому приховування даних може виконуватися в окремих фрагментах або частотних діапазонах. У запропонованому методі використовується

принцип мінімального впливу на сприйняття, зміни в молодших бітах семплів або у менш помітних частотах не впливають на якість прослуховування. Це дозволяє передавати приховану інформацію таким чином, що навіть при повторному кодуванні або обробці аудіо більшість прихованих даних зберігається. Акустичні властивості людського слуху, зокрема знижена чутливість до високих частот і короткочасних змін інтенсивності, роблять аудіо контейнером, який поєднує непомітність із достатньою прихованою ємністю.

Використання одночасно двох різних типів контейнерів дає змогу реалізувати справжній багатоканальний підхід, у якому дані розподіляються між зображенням та аудіофайлом відповідно до їхніх властивостей. Такий підхід суттєво ускладнює виявлення прихованого повідомлення, оскільки потенційний зловмисник повинен аналізувати не один, а одразу два різнорідні носії. Крім того, динамічний алгоритм розподілу дозволяє адаптивно визначати, яка частина інформації буде передана кожним каналом залежно від складності зображення, характеристик аудіосигналу, рівня шуму та інших параметрів. Це підвищує гнучкість методу та дозволяє мінімізувати ризик статистичних атак, що орієнтуються на аналіз окремого контейнера.

Для прикладу розподілу повідомлення між носіями розроблено Таблицю 2.3 де показано як повідомлення розміром 1000 біт ділиться між носіями у відсотковому співвідношенні.

Таблиця 2.3 – приклад розподілу.

Співвідношення		Початковий розмір повідомлення (біти)	Ємність каналів		Розподіл (біти)	
Зображення (канал 1)	Аудіо (канал 2)		Зображення (канал 1)	Аудіо (канал 2)	Зображення (канал 1)	Аудіо (канал 2)

100%	0%	1000	2000	Канал не обрано	1000	0
75%	25%	1000	1500	500	750	250
50%	50%	1000	2000	2000	500	500
25%	75%	1000	500	1500	250	750
0%	100%	1000	Канал не обрано	2000	0	1000

Таким чином, вибір зображення та аудіофайлу як базових каналів приховування інформації є науково обґрунтованим і відповідає вимогам до сучасних стеганографічних систем. Їх поєднання в межах динамічного багатоканального підходу дозволяє забезпечити високу непомітність, адаптивність, стійкість до атак і здатність до гнучкого перерозподілу навантаження між каналами відповідно до характеристик кожного носія.

2.3. Проектування алгоритму основних алгоритмів роботи вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

В даній роботу пропонується вдосконалити існуючий багатоканальний стеганографічний метод (Multi-image steganography) шляхом додавання можливості динамічного розподілу приховуваної інформації між каналами. Також пропонується вдосконалити алгоритм таким чином щоб каналами для приховування були файли різних типів, тобто до прикладу аудіо файл і зображення. На відмінну від наявного методу де каналами для приховування інформації слугують однотипні файли, до прикладу фото [39].

Виходячи з цього в даному підрозділі буде розроблено запропоновані основні алгоритми роботи системи вдосконаленого багатоканального

стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Для досягнення цієї мети першим пропонується розробити загальний алгоритм роботи вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації, в якому все покроково розписано.

Детальний, покроковий огляд загального алгоритму приховування інформації вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації:

Крок 1 - Початок.

Користувач запускає алгоритм відкривши програму;

Крок 2 - Введення повідомлення.

Користувач вводить повідомлення яке потрібно приховати в обрані носії;

Крок 3 - Повідомлення введено?

Крок 3.1 - Ні.

Якщо не введено повідомлення для приховування з`являється відповідна помилка яка виводиться користувачу;

Крок 4 - Вибір носіїв.

Крок 4.1 - Користувач обирає зображення як носій;

Крок 4.2 - Користувач обирає аудіо файл як носй;

Крок 5 - Носії обрано?

Відбувається аудит того чи обрано носії в які потрібно приховати інформацію;

Крок 5.1 - Ні.

Якщо не обрано хоча б 1 носій, алгоритм зупиняється і користувачу надсилається відповідна помилка;

Крок 6 - Введення пароля.

Користувач вводить пароль на основі якого буде відбуватися шифрування розподілених фрагментів введеного повідомлення.

Крок 7 - Пароль введено?

Крок 7.1 - Ні.

Якщо користувач не ввів пароль з`являється відповідна помилка і виводиться користувачу;

Крок 8 - Розподіл повідомлення між носіями.

Відбувається розподіл введеного повідомлення між обраними носіями, на основі параметрів самих носіїв та повідомлення.

Крок 9 - Шифрування частинами.

Відбувається шифрування введеного повідомлення на основі введеного пароля;

Крок 10 - Приховування інформації в носії.

Відбувається приховування кожної із частин інформації в кожен із обраних носіїв методом LSB.

Крок 11 - Формування вихідних стегано-файлів.

Відбувається формування стегано-файлів вже із прихованою інформацією всередині для можливості подальшого завантаження;

Крок 12 - Кінець.

Таким чином реалізується загальний покроковий алгоритм приховування для вдосконаленого багатоканального стеганографічного методу з динамічним розподілом інформації.

Також розроблено блок-схему даного алгоритму роботи системи, та схематично зображено за допомогою онлайн ресурсу Lucidchart (рис. 2.1) [40].

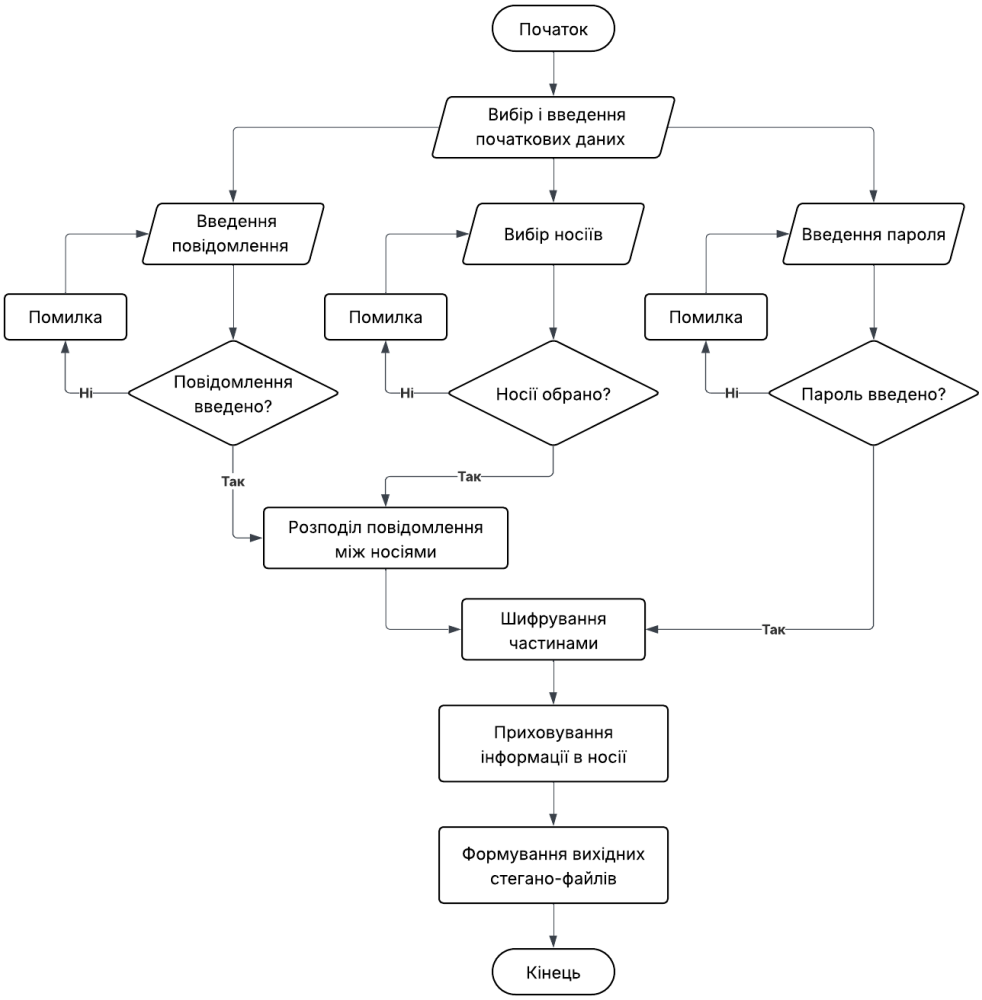


Рисунок 2.1 – блок-схема загального алгоритму приховування інформації у вдосконаленому багатоканальному стеганографічному методі з динамічним розподілом інформації.

Даний алгоритм включає і показує всі процеси які відбуваються в процесі приховування інформації.

Наступним було розроблено покроковий алгоритм динамічного розподілу інформації у вдосконаленому багатоканальному стеганографічному методі:

Крок 1 - Початок.

Користувач запускає алгоритм відкривши програму;

Крок 2 - Вибір носіїв.

Крок 2.1 - Користувач обирає зображення як носій;

Крок 2.2 - Користувач обирає аудіо файл як носй;

Крок 3 - Носії обрано?

Відбувається аудит того чи обрано носії в які потрібно приховати інформацію;

Крок 3.1 - Ні.

Якщо не обрано хоча б 1 носій, алгоритм зупиняється і користувачу надсилається відповідна помилка;

Крок 4 - Обчислення ємності носіїв.

Відбувається обчислення носіїв (скільки бітів інформації вміститься в кожен із них);

Крок 5 - Обчислення % співвідношення розподілу.

Відбувається обчислення носіїв у відсотковому співвідношення;

Крок 6 - Введення повідомлення для приховування.

Користувач вводить повідомлення яке потрібно приховати в обрані носії;

Крок 7 - Повідомлення введено?

Крок 7.1 - Ні.

Якщо не введено повідомлення для приховування з`являється відповідна помилка яка виводиться користувачу;

Крок 8 - Обчислення об`єму повідомлення.

Відбувається обчислення об`єму повідомлення яке було введено, для подальшого аналізу;

Крок 9 - Об`єм повідомлення більший за об`єм носіїв?

Крок 9.1 - Так.

Якщо об`єм введеного повідомлення більший за об`єм обраних носіїв з`являється відповідна помилка і виводиться користувачу;

Крок 10 - Розподіл повідомлення на частини.

Відбувається розподіл повідомлення на 2 частини, на основі обчисленого відсоткового співвідношення носіїв;

Крок 11 - Приховування в носії.

Відбувається приховування в обрані носії відносно попереднього розподілення.

Таким чином реалізований динамічний розподіл приховуваної інформації який залежить від даних самих носіїв.

Також розроблено блок-схему алгоритму динамічного розподілу інформації у вдосконаленому багатоканальному стеганографічному методі (рис. 2.2).

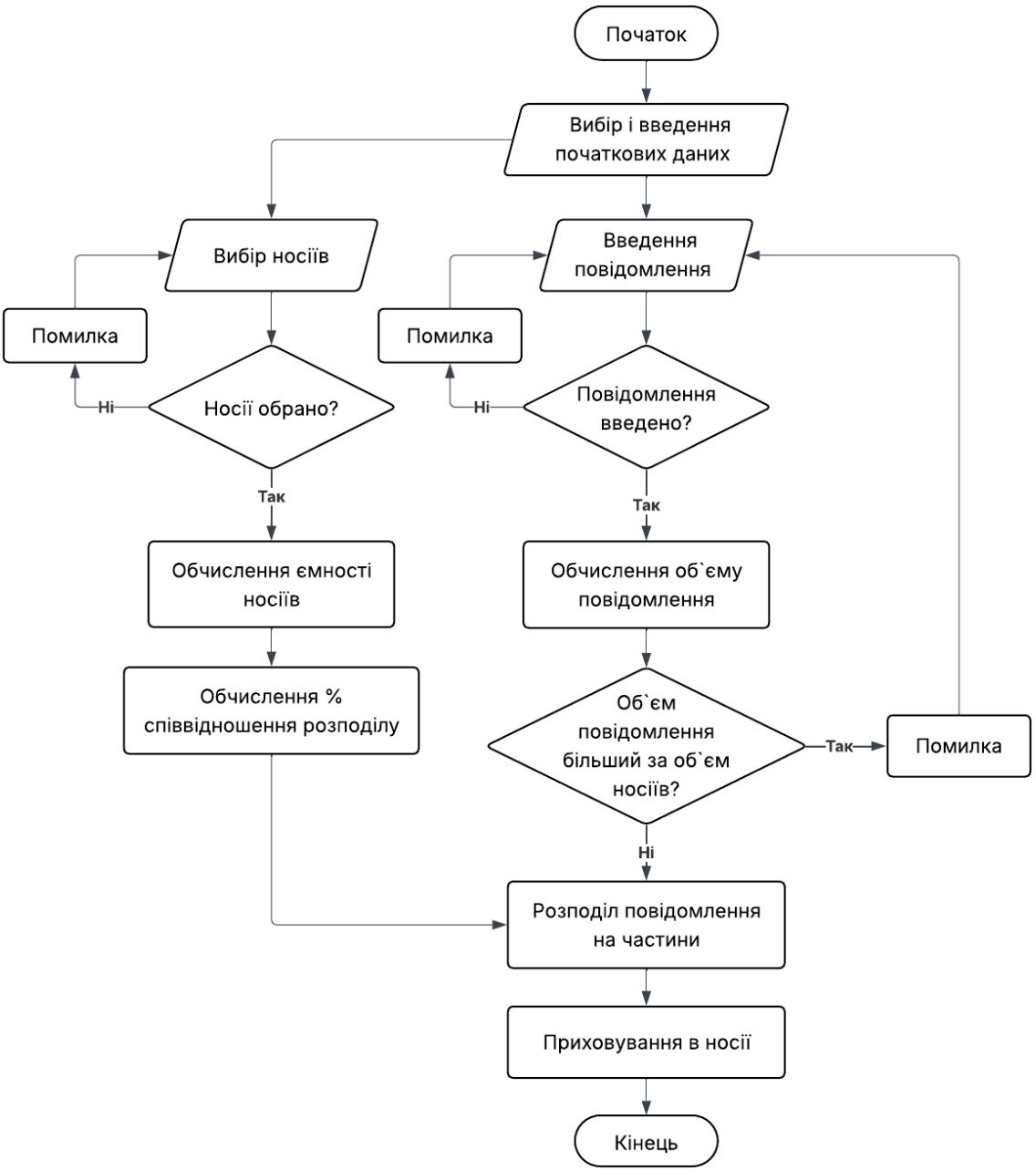


Рисунок 2.1 – блок-схема алгоритму динамічного розподілу інформації у вдосконаленому багатоканальному стеганографічному методі з динамічним розподілом інформації.

Алгоритм описує загальні ключові кроки в процесі динамічного розподілу приховуваної інформації.

Наступним розроблено алгоритм шифрування введеного користувачем повідомлення для приховування [41].

Крок 1 - Початок.

Користувач запускає алгоритм відкривши програму;

Крок 2 - Введення повідомлення.

Користувач вводить повідомлення яке потрібно приховати в обрані носії;

Крок 3 - Повідомлення введено?

Крок 3.1 - Ні.

Якщо не введено повідомлення для приховування з`являється відповідна помилка яка виводиться користувачу;

Крок 4 - Перетворення в байти.

Якщо користувач ввів повідомлення воно перетворюється в байти для подальшої обробки.

Крок 5 - Введення пароля.

Користувач вводить пароль на основі якого буде відбуватися шифрування розподілених фрагментів введеного повідомлення.

Крок 6 - Пароль введено?

Крок 6.1 - Ні.

Якщо користувач не ввів пароль з`являється відповідна помилка і виводиться користувачу;

Крок 7 - Генерація ключа.

Відбувається генерація ключа на основі введеного пароля для шифрування введеного повідомлення в подальшому;

Крок 8 - Розподіл повідомлення між носіями.

На даному етапі повідомлення ділиться між обраними носіями, на основі властивостей обраних носіїв.

Крок 9 - Шифрування частинами.

Відбувається шифрування кожної частини розподіленого повідомлення на основі введеного повідомлення і згенерованого ключа.

Крок 10 - Формування шифрованого блоку.

На даному етапі відбувається формування шифрованого блоку для подальшого приховування в кожен із каналів.

Крок 11 - Кінець.

Таким чином відбувається процес шифрування введеного повідомлення вдосконаленого багатоканального стеганографічного методу з динамічним розподілом інформації.

Також розроблено блок-схему даного алгоритму шифрування (рис. 2.3).

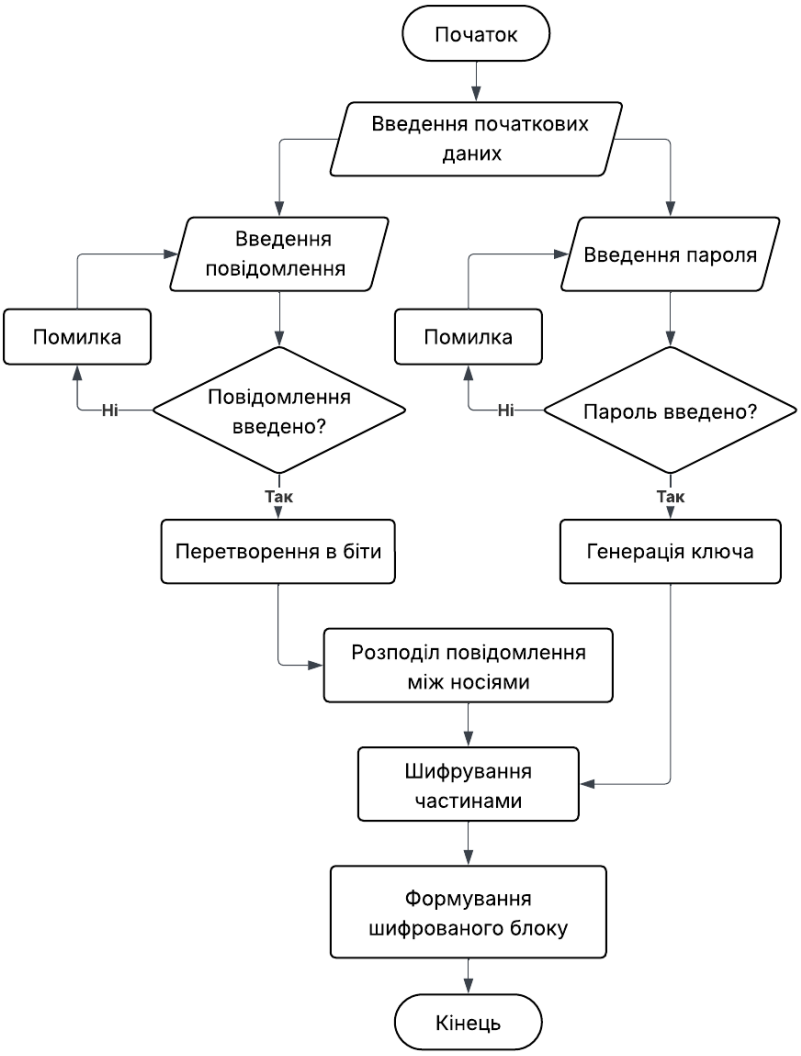


Рисунок 2.1 – блок-схема алгоритму шифрування розподіленої інформації у вдосконаленому багатоканальному стеганографічному методі з динамічним розподілом інформації.

Алгоритм забезпечує безпеку під час приховування інформації в носії, додатково шифруючи інформацію перед приховуванням.

Таким чином було розроблено основні із алгоритмів роботи системи, та побудованого їх блок-схеми.

2.4. Висновки до розділу 2.

В даному розділі було здійснено розробку і проектування ключових алгоритмів роботи вдосконаленого багатоканального стеганографічного методу з динамічним розподілом інформації, для подальшої програмної реалізації. Було розроблено наступні алгоритми роботи: загальний алгоритм роботи під час приховування інформації, алгоритм динамічного розподілу інформації між носіями на основі даних і властивостей носіїв та алгоритм шифрування розділених частин повідомлення перед приховуванням для кращого захисту інформації на основі введеного користувачем пароля та згенерованого ключа на основі пароля.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВДОСКОНАЛЕНОГО БАГАТОКАНАЛЬНОГО СТЕГАНОГРАФІЧНОГО МЕТОДУ З ВИКОРИСТАННЯМ ДОНАМІЧНОГО РОЗПОДІЛУ ІНФОРМАЦІЇ

В даному розділі буде обрано і аргументовано вибір мови програмування та середовища розробки для програмної резлізації вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації. Також буде здійснено саму програмну реалізацію вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації, та проведено його тестування з окресленням переваг вдосконалення над існуючими рішеннями. Буде зроблено висновки.

3.1. Обґрунтування вибору мови програмування та середовища розробки.

Після завершення проектування алгоритмів, блок схем та інтерфейсу вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації, необхідно визначити за допомогою якої мови програмування та середовища розробки буде програмно реалізовано вдосконалений багатоканальний стеганографічний метод з використанням динамічного розподілу прихованої інформації.

Для програмної реалізації вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації, прийнято рішення використовувати мову програмування JavaScript [42].

JavaScript – це одна з найпопулярніших, об’єктно-орієнтованих мов програмування в світі, яка підтримується абсолютно всіма існуючими

браузерами, так як побудована на рушії V8. JavaScript є досить гнучкою і універсальною мовою програмування [43].

Під час розробки вдосконаленого багатоканального стеганографічного методу з динамічним розподілом інформації було обрано середовище розробки Visual Studio Code (VS Code). Такий вибір зумовлений поєднанням високої продуктивності середовища, зручності використання, широких можливостей налаштування та підтримки сучасних мов програмування і технологій. Visual Studio Code є дуже легким, але водночас потужним інструментом, який забезпечує ефективний процес розробки як для невеликих проєктів, так і для комплексних систем [44].

Однією з головних переваг VS Code є його кросплатформність, що дозволяє працювати з ним на операційних системах Windows, Linux і macOS без втрати функціональності. Це робить середовище зручним для командної роботи або розробників, які використовують різні системи. Крім того, VS Code підтримує широкий спектр мов програмування, зокрема JavaScript яку обрано для розробки, Python, C++, дозволяє легко працювати з структурою проекту за допомогою HTML та із стилями за допомогою CSS це забезпечує універсальність під час написання коду [45].

Середовище Visual Studio Code вирізняється гнучкістю налаштування адже користувач може розширювати його функціонал за допомогою численних плагінів і розширень, доступних у вбудованому маркетплейсі. Це дозволяє адаптувати середовище до конкретних потреб проєкту та тим самим підвищити ефективність розробки. Також VS Code має зручну інтеграцію з системами контролю версій, зокрема Git, що спрощує спільну роботу над кодом і відстеження змін у проєкті [46].

Не менш важливою перевагою є зручний інтерфейс і підтримка інтелектуального автодоповнення коду (IntelliSense), яке допомагає зменшити кількість синтаксичних помилок і прискорює написання програмного коду. Крім того, Visual Studio Code підтримує налагодження

(debugging) у реальному часі, що дозволяє швидко виявляти й виправляти помилки, також Visual Studio Code підтримує багато мов власного інтерфейсу.

Важливим моментом є те що активна спільнота розробників і регулярні оновлення від компанії Microsoft гарантують стабільність роботи середовища, постійне вдосконалення його функцій і підтримку найновіших світових технологій.

Отже, вибір Visual Studio Code як середовища розробки є цілком обґрунтованим, оскільки воно забезпечує зручність, гнучкість, стабільність і високу продуктивність роботи, що відповідає вимогам сучасного процесу створення будь якого програмного забезпечення.

3.2. Програмна реалізація вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Перед початком програмної реалізації вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації, створено файл stego.html для програмної реалізації розробленого в попередньому розділі інтерфейсу та прописано відповідний код [47].

```
<!doctype html>
<html lang="uk">
<head>
 <meta charset="utf-8" />
 <meta name="viewport" content="width=device-width,initial-scale=1" />
 <title>Вдосконалений багатоканальний стеганографічний метод з динамічним розподілом інформації</title>
 <link rel="stylesheet" href="style.css">
</head>
<body>
 <h1>Вдосконалений багатоканальний стеганографічний метод з динамічним розподілом інформації</h1>

 <div class="card upload" id="imageUpload">
  <label>1) Завантажити зображення (PNG або JPG)</label>
  <div class="upload-area" id="imageArea">
   <input id="fileImage" type="file" accept="image/*">
```

```
            <span>Натисніть, щоб вибрати зображення</span>
           </div>
           <canvas id="canvas" width="400" height="300" style="display:block"></canvas>
          </div>

          <div class="card upload" id="audioUpload">
           <label>2) Завантажити аудіофайл</label>
           <div class="upload-area" id="audioArea">
            <input id="fileAudio" type="file" accept="audio/*">
            <span>Натисніть, щоб вибрати аудіо</span>
           </div>
           <audio id="audioPreview" controls style="width:100%; margin-top:8px;
display:none;"></audio>
           <div id="audioInfo" class="muted"></div>
          </div>

          <div class="card">
           <label>3) Пароль (ключ для шифрування)</label>
           <input id="password" type="password" placeholder="Введіть пароль...">

           <label>4) Повідомлення для приховування</label>
           <textarea id="message" placeholder="Введіть текст..."></textarea>

           <div class="row">
            <button id="encodeBtn">Приховати</button>
            <button id="decodeBtn">Видобути</button>
            <button id="downloadImageBtn" disabled>Завантажити стего-зображення</button>
            <button id="downloadAudioBtn" disabled>Завантажити стего-аудіо</button>
           </div>

           <div id="info" class="muted" style="margin-top:10px"></div>
          </div>
          <script src="script.js"></script>
         </body>
         </html>
```

Після чого створено файл index.js для написання логіки вдосконаленого методу. Далі в цьому файлі було прописано всю логіку вдосконаленого багатоканального стеганографічного методу з динамічним розподілом інформації [48].

Спочатку отримано дані з полів введення з html та оголошено змінні, для подальшої роботи з ними.

```
const fileImage = document.getElementById('fileImage');
const fileAudio = document.getElementById('fileAudio');
const canvas = document.getElementById('canvas');
const ctx = canvas.getContext('2d');
const messageEl = document.getElementById('message');
const encodeBtn = document.getElementById('encodeBtn');
const decodeBtn = document.getElementById('decodeBtn');
const downloadImageBtn = document.getElementById('downloadImageBtn');
const downloadAudioBtn = document.getElementById('downloadAudioBtn');
```

```
const info = document.getElementById('info');
const audioInfo = document.getElementById('audioInfo');
const passwordEl = document.getElementById('password');

let originalImage = null;
let originalAudioBuffer = null;
let originalAudioFile = null;
let audioSampleRate = 44100;
let lastImageBitsUsed = 0;
let lastAudioBitsUsed = 0;
```

Після цього прописано функцію яка перетворює текст який користувач хоче приховати в послідовність бітів для подального приховування методом LSB [49].

```
function bitsFromNumber(n, bitsCount) {
const out = [];
for(let i=bitsCount-1;i>=0;i--) out.push((n >> i) & 1);
return out;
}
function bitsFromBytes(byteArray) {
const out = [];
for (let b of byteArray) for (let i = 7; i >= 0; i--) out.push((b >> i) & 1);
return out;
}
function bytesFromBits(bits) {
const bytes = [];
for (let i = 0; i < bits.length; i += 8) {
let val = 0;
for (let j = 0; j < 8 && i+j < bits.length; j++) val = (val << 1) | bits[i+j];
bytes.push(val);
}
return new Uint8Array(bytes);
}
```

Наступним було прописано функцію deriveKey() яка відповідає за шифрування бажаного повідомлення (послідовність бітів отриманих в попередній функції). Спершу введений користувачем пароль перетворюється у масив байтів та створюється криптографічний AES ключ. На основі цього ключа шифрується саме повідомлення. Також реалізована можливість зворотного дешифрування до початкового вигляду повідомлення.

```
async function deriveKey(password, salt) {
const enc = new TextEncoder();
const pwKey = await crypto.subtle.importKey('raw', enc.encode(password), {name: 'PBKDF2'},
false, ['deriveKey']);
return crypto.subtle.deriveKey(
{ name: 'PBKDF2', salt: salt, iterations: 200000, hash: 'SHA-256' },
```

```
pwKey,
{ name: 'AES-GCM', length: 256 },
false,
['encrypt','decrypt']
);
}
async function encryptBytes(password, plainBytes) {
const salt = crypto.getRandomValues(new Uint8Array(16));
const iv = crypto.getRandomValues(new Uint8Array(12));
const key = await deriveKey(password, salt);
const ct = await crypto.subtle.encrypt({ name: 'AES-GCM', iv: iv }, key, plainBytes);
return { salt, iv, ciphertext: new Uint8Array(ct) };
}
async function decryptBytes(password, salt, iv, ciphertext) {
const key = await deriveKey(password, salt);
try {
const pt = await crypto.subtle.decrypt({ name: 'AES-GCM', iv: iv }, key, ciphertext);
return new Uint8Array(pt);
} catch (e) {
return null;
}
}
```

Після чого було прописано функцію () яка відповідає за перетворення аудіо з окремих каналів у один інтерлівований масив Int16, готовий для запису у WAV-файл або вставки прихованих даних.

```
function interleaveChannels(channels) {
const length = channels[0].length;
const numChannels = channels.length;
const out = new Int16Array(length * numChannels);
for (let i = 0; i < length; i++) {
for (let ch = 0; ch < numChannels; ch++) {
let s = channels[ch][i];
s = Math.max(-1, Math.min(1, s));
out[i * numChannels + ch] = Math.round(s * 32767);
}
}
return out;
}
```

Наступним було прописано фрагмент коду який відповідає за створення WAV-файлу з масиву аудіосемплів, формуючи правильний заголовок і дані, щоб файл можна було програти у будь-якому аудіоплеєрі.

```
function writeWav(int16Samples, sampleRate, numChannels) {
const bytesPerSample = 2;
const blockAlign = numChannels * bytesPerSample;
const byteRate = sampleRate * blockAlign;
const dataSize = int16Samples.length * bytesPerSample;
const buffer = new ArrayBuffer(44 + dataSize);
const dv = new DataView(buffer);
let p = 0;
```

```
function writeString(s) { for (let i=0;i<s.length;i++) dv.setUint8(p++, s.charCodeAt(i)); }
writeString('RIFF'); dv.setUint32(p, 36 + dataSize, true); p += 4; writeString('WAVE');
writeString('fmt '); dv.setUint32(p,16,true); p+=4; dv.setUint16(p,1,true); p+=2;
dv.setUint16(p,numChannels,true); p+=2; dv.setUint32(p,sampleRate,true); p+=4;
dv.setUint32(p,byteRate,true); p+=4; dv.setUint16(p,blockAlign,true); p+=2; dv.setUint16(p,16,true); p+=2;
writeString('data'); dv.setUint32(p,dataSize,true); p+=4;
for (let i=0;i<int16Samples.length;i++,p+=2) dv.setInt16(p, int16Samples[i], true);
return new Blob([buffer], {type: 'audio/wav'});
}
```

Наступним було приписано функцію splitPlaintext() яка відповідає да сам динамічний розподіл приховуваної інформації у вдосконаленому багатоканальному стеганографічному методі з використанням динамічного розподілу прихованої інформації.

Спочатку розраховується загальна ємність обраних користувачем носіїв для приховування (фото та аудіо файл), тобто визначається скільки бітів можна приховати в кожен з носіїв, та визначається чи взагалі є куди приховувати (чи носії обрано). Після чого відбувається пропорційний розподіл між носіями з визначенням кількості біт а байт для кожного з них, на основі отриманих раніше біт носіїв і приховуваної інформації.

```
function splitPlaintext(plainBytes, imageBits, audioBits) {
  const totalCarrierBits = imageBits + audioBits;
  if (totalCarrierBits === 0) return {firstChunk: new Uint8Array(0), secondChunk: new
Uint8Array(0)};
  const audioRatio = audioBits / totalCarrierBits;
  const totalBits = plainBytes.length * 8;
  const audioBitsToUse = Math.round(totalBits * audioRatio);
  const audioBytesCount = Math.floor(audioBitsToUse / 8);
  const imageBytesCount = plainBytes.length - audioBytesCount;
  const firstChunk = plainBytes.slice(0, imageBytesCount);
  const secondChunk = plainBytes.slice(imageBytesCount);
  return {firstChunk, secondChunk};
}
```

Наступним було прописано код який відповідає за завантаження зображення в пам’ять, промальовування в canvas та отримання його пікселів для подальшої обробки та приховування.

```
fileImage.addEventListener('change', e => {
const f = e.target.files[0];
if(!f) return;
const img = new Image();
img.onload = () => {
canvas.width = img.width;
canvas.height = img.height;
```

```
ctx.drawImage(img,0,0);
originalImage = img;
downloadImageBtn.disabled = true;
updateInfo();
};
img.src = URL.createObjectURL(f);
});
```

Після чого було прописано алгоритм функції embedToImage() яка приховує розподілену раніше частину приховуваного тексту в фото стеганографіним методом LSB записуючи біти повідомлення в найменш значущі біти каналів R, G і B кожного пікселя.

```
function embedToImage(imageData, bytes) {
const data = imageData.data;
const w = imageData.width, h = imageData.height;
const totalBits = w*h*3;
const neededBits = (4 + bytes.length) * 8;
if (neededBits > totalBits) throw new Error('Image capacity too small for given chunk');
const lenBits = bitsFromNumber(bytes.length >>> 0, 32);
const msgBits = bitsFromBytes(bytes);
const bits = lenBits.concat(msgBits);
let bitIdx = 0;
for (let px = 0; px < w*h && bitIdx < bits.length; px++) {
const base = px*4;
for (let c = 0; c < 3 && bitIdx < bits.length; c++) {
data[base+c] = (data[base+c] & 0xFE) | bits[bitIdx++];
}
}
lastImageBitsUsed = bitIdx;
}
```

Після чого було приписано частину коду який відповідає за подібну підготовку завантаженого аудіо носія до приховування в нього інформації, перетворюючи його формат у зручний для обробки. Частина коду відповідає за завантаження аудіофайлу користувачем, його декодування у формат для Web Audio API та відображення інформації про аудіо (кількість семплів, каналів та частоту дискретизації). Фрагмент коду також зберігає аудіодані для подальшого використання у стеганографії.

```
const audioPreview = document.getElementById('audioPreview');

const audioEl = audioPreview.querySelector('audio');

fileAudio.addEventListener('change', async e => {
  const f = e.target.files[0];
  if (!f) return;
  originalAudioFile = f;
```

```
        audioEl.src = URL.createObjectURL(f);
        audioEl.load();
        audioEl.play();
        audioPreview.style.display = 'block';

        const ab = await f.arrayBuffer();
        try {
          const buf = await audioCtx.decodeAudioData(ab.slice(0));
          originalAudioBuffer = buf;
          audioSampleRate = buf.sampleRate;
          audioInfo.textContent = `Audio: ${buf.length} samples, ${buf.numberOfChannels} channels,
${buf.sampleRate} Hz.`;
          downloadAudioBtn.disabled = true;
          updateInfo();
        } catch (err) {
          audioInfo.textContent = 'Не вдалося декодувати аудіо.';
          originalAudioBuffer = null;
          updateInfo();
        }
      });
```

Наступною функцією було створено функція embedToAudio() как реалізує алгоритм приховування тексту в аудіо файл також стеганографічним методом LSB. Алгоритм функції перетворює канали в Int16, об’єднує їх у один потік (interleaved), записує біти повідомлення в найменш значущі біти семплів і повертає як інтегровані семпли, так і оновлені плаваючі канали для подальшого використання.

```
function embedToAudio(audioBuffer, bytes) {
const numChannels = audioBuffer.numberOfChannels;
const length = audioBuffer.length;
const totalBits = length * numChannels;
const neededBits = (4 + bytes.length) * 8;
if (neededBits > totalBits) throw new Error('Audio capacity too small for given chunk');

const channels = [];
for (let ch = 0; ch < numChannels; ch++) {
const floatArr = audioBuffer.getChannelData(ch);
const intArr = new Int16Array(length);
for (let i=0;i<length;i++) intArr[i] = Math.round(Math.max(-1, Math.min(1, floatArr[i])) * 32767);
channels.push(intArr);
}

const interleaved = new Int16Array(length * numChannels);
for (let i = 0; i < length; i++) for (let ch=0; ch<numChannels; ch++)
interleaved[i*numChannels+ch] = channels[ch][i];

const lenBits = bitsFromNumber(bytes.length >>> 0, 32);
const msgBits = bitsFromBytes(bytes);
const bits = lenBits.concat(msgBits);
```

```
for (let i = 0; i < bits.length; i++) {
interleaved[i] = (interleaved[i] & 0xFFFE) | bits[i];
}
lastAudioBitsUsed = bits.length;
```

Для отримання масиву каналів з аудіо файлу прописано наступну частину коду, де для кожного каналу створюється масив Float32, в якому семпли нормалізуються з цілочисельного діапазону Int16 (-32767…32767) у плаваючі числа (-1…1).

```
const outChannels = [];
const samplesPerChannel = interleaved.length / numChannels;
for (let ch=0; ch<numChannels; ch++) {
const floatArr = new Float32Array(samplesPerChannel);
for (let i=0;i<samplesPerChannel;i++) {
floatArr[i] = interleaved[i*numChannels + ch] / 32767;
}
outChannels.push(floatArr);
}

return {int16Interleaved: interleaved, channelsFloat: outChannels};
}
```

На даному етапі вже частково реалізований функціонал приховування інформації в фото та аудіо файл з динамічним розподілом. Але для того щоб це працювало в двосторонньому напрямку, тобто дані можна було як приховувати в носії так і добувати приховану раніше інформацію чим методом, було розроблено цей функціонал в зворотному напрямку.

Функція extractFromImage() дістає з фото біти прихованої інформації, шляхом читання бітів з найменш значущих бітів кольорових каналів та визначає довжину повідомлення, після чого відновлює оригінальні байти, навіть якщо дані частково обрізані, в результаті отримуємо зашифровану частину повідомлення яка була прихована в фото носій.

```
function extractFromImage(imageData) {
const data = imageData.data;
const w = imageData.width, h = imageData.height;
const bits = [];
for (let px = 0; px < w*h; px++) {
const base = px*4;
for (let c = 0; c < 3; c++) bits.push(data[base+c] & 1);
}

let len = 0;
if (bits.length < 32) return new Uint8Array(0);
```

```
for (let i = 0; i < 32; i++) len = (len << 1) | bits[i];
const totalNeeded = 32 + len*8;
if (bits.length < totalNeeded) {

const availableBits = Math.max(0, bits.length - 32);
const partialBytes = Math.floor(availableBits / 8);
const msgBits = bits.slice(32, 32 + partialBytes*8);
return bytesFromBits(msgBits);
}
const msgBits = bits.slice(32, 32 + len*8);
return bytesFromBits(msgBits);
}
```

Наступною була приписана функція extractFromAudioBuffer () яка виконує туж саму задачу що і попередня функція але за витягує приховані дані з аудіо. Алгоритм функції перетворює канали аудіо у цілі числа, читає найменш значущі біти для відновлення довжини повідомлення, а потім повертає відновлені байти, навіть якщо дані частково обрізані, таким чином ми отримуємо зашифрований фрагмент прихованої інформації в аудіо файл.

```
function extractFromAudioBuffer(audioBuffer) {
const numChannels = audioBuffer.numberOfChannels;
const length = audioBuffer.length;
const interleaved = new Int16Array(length * numChannels);
for (let ch=0; ch<numChannels; ch++) {
const floatArr = audioBuffer.getChannelData(ch);
for (let i=0;i<length;i++) interleaved[i*numChannels+ch] = Math.round(Math.max(-1, Math.min(1,
floatArr[i])) * 32767);
}
const bits = [];
for (let i=0;i<interleaved.length;i++) bits.push(interleaved[i] & 1);
if (bits.length < 32) return new Uint8Array(0);
let len = 0;
for (let i = 0; i < 32; i++) len = (len << 1) | bits[i];
const totalNeeded = 32 + len*8;
if (bits.length < totalNeeded) {
const availableBits = Math.max(0, bits.length - 32);
const partialBytes = Math.floor(availableBits / 8);
const msgBits = bits.slice(32, 32 + partialBytes*8);
return bytesFromBits(msgBits);
}
const msgBits = bits.slice(32, 32 + len*8);
return bytesFromBits(msgBits);
}
```

Наступним було прописано код який відповідає за кодування повідомлення користувача у доступні носії (зображення та аудіо). Спершу алгоритм бере текст із поля введення та пароль і перевіряє, чи не порожні вони. Потім текст перетворюється в масив байтів, а також обчислюється

кількість бітів, які можна приховати у зображенні та аудіо. Повідомлення ділиться на дві частини відповідно до пропорцій доступних бітів: перша частина йде для зображення, друга для аудіо. Кожна частина шифрується за допомогою AES-GCM з використанням пароля, а результати включають salt, iv та зашифровані дані.

Після цього закодовані біти вставляються у відповідні носії: у пікселі зображення або у аудіосемпли, при цьому для аудіо ще формується WAV-файл. Після успішного кодування кнопки завантаження стего-файлів стають активними, а на екрані відображається інформація про використані біти та розподіл повідомлення між носіями (у відсотковому співвідношенні). Якщо носії відсутні або виникає помилка під час процесу, користувач отримує повідомлення про помилку.

```
encodeBtn.addEventListener('click', async () => {
try {
const text = messageEl.value;
if (!text) return alert('Введіть повідомлення.');
const password = passwordEl.value;
if (!password) return alert('Введіть пароль.');

const plainBytes = new TextEncoder().encode(text);

const w = canvas.width, h = canvas.height;
const imageBits = originalImage ? w*h*3 : 0;
const audioBits = originalAudioBuffer ? originalAudioBuffer.length *
originalAudioBuffer.numberOfChannels : 0;

const {firstChunk: firstPlain, secondChunk: secondPlain} = splitPlaintext(plainBytes, imageBits,
audioBits);

let imgEmbedBytes = new Uint8Array(0);
let audioEmbedBytes = new Uint8Array(0);

if (firstPlain.length > 0 && originalImage) {
const {salt, iv, ciphertext} = await encryptBytes(password, firstPlain);
imgEmbedBytes = new Uint8Array(salt.length + iv.length + ciphertext.length);
imgEmbedBytes.set(salt, 0);
imgEmbedBytes.set(iv, salt.length);
imgEmbedBytes.set(ciphertext, salt.length + iv.length);
}

if (secondPlain.length > 0 && originalAudioBuffer) {
const {salt, iv, ciphertext} = await encryptBytes(password, secondPlain);
audioEmbedBytes = new Uint8Array(salt.length + iv.length + ciphertext.length);
audioEmbedBytes.set(salt, 0);
audioEmbedBytes.set(iv, salt.length);
```

```
audioEmbedBytes.set(ciphertext, salt.length + iv.length);
}

if (originalImage && imgEmbedBytes.length > 0) {
const imageData = ctx.getImageData(0,0,w,h);
embedToImage(imageData, imgEmbedBytes);
ctx.putImageData(imageData,0,0);
downloadImageBtn.disabled = false;
}

if (originalAudioBuffer && audioEmbedBytes.length > 0) {
const {int16Interleaved} = embedToAudio(originalAudioBuffer, audioEmbedBytes);
const wavBlob = writeWav(int16Interleaved, audioSampleRate,
originalAudioBuffer.numberOfChannels);
window._lastStegoAudioBlob = wavBlob;
downloadAudioBtn.disabled = false;
}

if ((!originalImage || imgEmbedBytes.length === 0) && (!originalAudioBuffer ||
audioEmbedBytes.length === 0)) {
return alert('Немає доступних носіїв або повідомлення розподілено у нуль байт для
доступних носіїв.');
}

alert('Кодування завершено. Якщо носій доступний, збережіть стего-файли.');
lastImageBitsUsed = lastImageBitsUsed || 0;
lastAudioBitsUsed = lastAudioBitsUsed || 0;
updateInfo(plainBytes.length);
} catch (err) {
console.error(err);
alert('Помилка: ' + (err && err.message ? err.message : err));
}
});
```

Для того щоб після приховування інформації користувач міг завантажити носії (фото і аудіо файли), було прописано наступний код.

```
downloadImageBtn.addEventListener('click', () => {
canvas.toBlob(blob => {
const a = document.createElement('a');
const url = URL.createObjectURL(blob);
a.href = url; a.download = 'stego_image.png'; a.click(); URL.revokeObjectURL(url);
}, 'image/png');
});

downloadAudioBtn.addEventListener('click', () => {
const blob = window._lastStegoAudioBlob;
if (!blob) return alert('Немає стего-аудіо.');
const a = document.createElement('a');
const url = URL.createObjectURL(blob);
a.href = url; a.download = 'stego_audio.wav'; a.click(); URL.revokeObjectURL(url);
});
```

Для того щоб вдосконалений багатоканальний стеганографічний метод з динамічним розподілом інформації був повноцінним, і працював в два боки (був двостороннім), тобто міг як приховувати інформацію так і витягувати її з файлів де вона вже прихована, було прописано наступний код.

```
decodeBtn.addEventListener('click', async () => {
try {
const password = passwordEl.value;
if (!password) return alert('Введіть пароль для розкодування.');

let imagePartBytes = new Uint8Array(0);
let audioPartBytes = new Uint8Array(0);

if (originalImage) {
const w = canvas.width, h = canvas.height;
const imageData = ctx.getImageData(0,0,w,h);
imagePartBytes = extractFromImage(imageData);
}

if (originalAudioBuffer) {
audioPartBytes = extractFromAudioBuffer(originalAudioBuffer);
}

let partsText = [];

async function tryDecryptPart(bytes) {
if (!bytes || bytes.length < 28) return null;
const salt = bytes.slice(0,16);
const iv = bytes.slice(16,28);
const ciphertext = bytes.slice(28);
const dec = await decryptBytes(password, salt, iv, ciphertext);
if (dec === null) return null;
return new TextDecoder().decode(dec);
}

const imgText = await tryDecryptPart(imagePartBytes);
const audText = await tryDecryptPart(audioPartBytes);

if (imgText !== null) partsText.push(imgText);
if (audText !== null) partsText.push(audText);

if (partsText.length === 0) {
const hadAny = (imagePartBytes && imagePartBytes.length > 0) || (audioPartBytes &&
audioPartBytes.length > 0);
if (hadAny) return alert('Наявні дані знайдено, але розшифрування не вдалося
(неправильний пароль або урізані дані).');
return alert('Немає даних для декодування у завантажених файлах.');
}

const finalText = partsText.join('');
messageEl.value = finalText;
alert('Декодування завершено. Показано доступні частини повідомлення.');
```

```
} catch (err) {
console.error(err);
alert('Помилка декодування: ' + (err && err.message ? err.message : err));
}
```

Для кращого розуміння і контроль контролю користувачем процесу приховування інформації вдосконаленим багатоканальним стеганографічним методом з динамічним розподілом інформації було додано можливіть вивведення основних метрик в процесі роботи алгоритму, а саме: місткість завантаженого зображення та аудіо для приховування даних, також відображає розмір і співвідношення розподілу прихованого повідомлення між зображенням і аудіо файлом.

```
function updateInfo(payloadPlainLength = 0) {
const w = canvas.width, h = canvas.height;
const imageBitsCapacity = originalImage ? w * h * 3 : 0;
const audioBitsCapacity = originalAudioBuffer ? originalAudioBuffer.length *
originalAudioBuffer.numberOfChannels : 0;
let infoText = ` Image capacity: ${originalImage ? ` ${w}×${h}px —
${Math.floor(imageBitsCapacity/8)} байт` : '— не завантажено'}.\n`;
infoText += ` Audio capacity: ${originalAudioBuffer ? ` ${originalAudioBuffer.length} сэмплів ×
${originalAudioBuffer.numberOfChannels} каналів — ${Math.floor(audioBitsCapacity/8)} байт` : '— не
завантажено'}.\n`;
if (payloadPlainLength > 0) {
const totalBitsUsed = lastImageBitsUsed + lastAudioBitsUsed;
const imgPercent = totalBitsUsed === 0 ? 0 :
Math.round((lastImageBitsUsed/totalBitsUsed)*100);
const audPercent = totalBitsUsed === 0 ? 0 :
Math.round((lastAudioBitsUsed/totalBitsUsed)*100);
infoText += ` \nПовідомлення (plain) байт: ${payloadPlainLength}. Приховано бітів: Image =
${lastImageBitsUsed}, Audio = ${lastAudioBitsUsed}\nРозподіл: Image ${imgPercent}%, Audio
${audPercent}%`;
} else {
infoText += ` \nВведіть повідомлення і натисніть "Закодувати".`;
}
info.textContent = infoText;
}
```

Таким чином було програмно реалізовано вдосконалений багатоканальний стеганографічний метод з використанням динамічного розподілу прихованої інформації. Під час розробки було використано раніше спроектовані алгоритми роботи та користувацький інтерфейс.

Після чого на заключному етапі було створено файл style.css в якому прописано стилі у відповідності до спроектованого інтерфейсу [45; 51].

3.3. Тестування вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

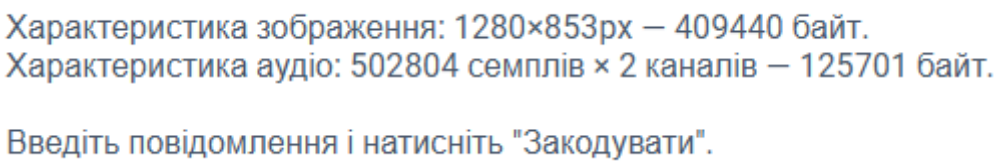
Наступним етапом в програмній реалізації вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації є тестування його працездатності.

Для цього було протестовано програмно реалізований вдосконалений багатоканальний стеганографічний метод з використанням динамічного розподілу прихованої інформації, в наступних режимах роботи:

- Тестування на продуктивність та масштабованість;
- Тестування динамічного розподілу;
- Тестування цілісності повідомлення після приховування та отримання у зворотньому порядку;
- Оцінка якості носіїв після вбудовування;
- Тестування криптографічної стійкості;
- Можливість одноканального приховування.

Для початку проведено тестування продуктивності та маштабованості, так як однією з переваг вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації є те що він значно збільшує об`єм інформації яку можна приховати.

Для цього було обрано носії для приховування: (фото 1280 на 853 пікселів), (аудіо тривалістю 00:06 хв) (рис. 3.1).



Зображення 3.1 - характеристика обраних носіїв (зображення та аудіо файлу).

Для приховування було обрано текст розміром 206328 символів, та проведено його приховування. В результаті даного тестування повідомлення було успішно приховане в обрані носії (рис. 3.2).

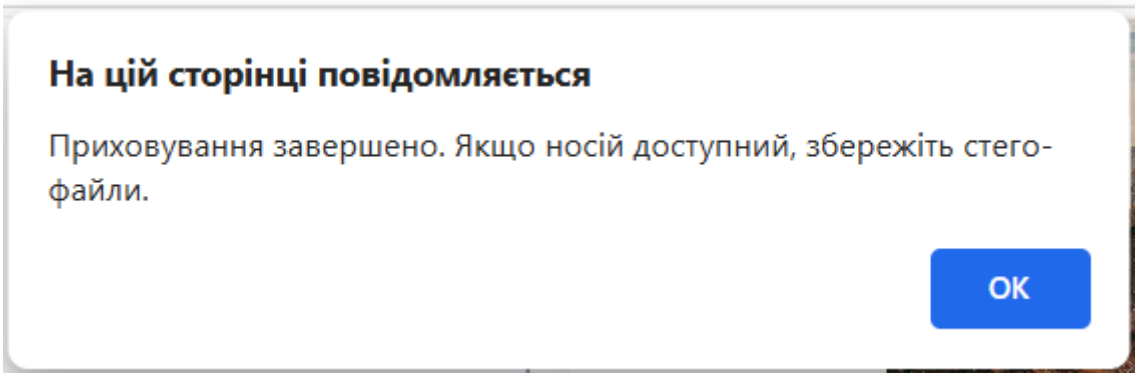
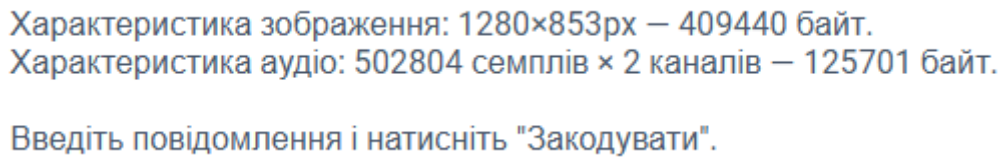


Рисунок 3.2 – результат тестування продуктивності та маштабованості, програмно реалізованого вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

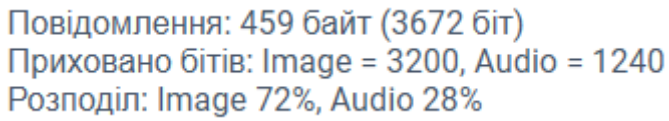
Наступним було проведено тестування динамічного розподілу інформації, так як це одне із ключових вдосконалень багатоканального стеганографічного методу. Для цього було обрано один і той самий текст для приховування та один і той самий носій, при чому інший під час тесту змінювався.

Тестування відбувається в 2 етапи. На першому етапі тестування обрано повідомлення для приховування об`ємом 459 символи, фото носій розміром 1280 на 853 пікселі та аудіо носій тривалістю 00:10 хв (рис. 3.3).



Зображення 3.3 - характеристика обраних носіїв (зображення та аудіо файлу).

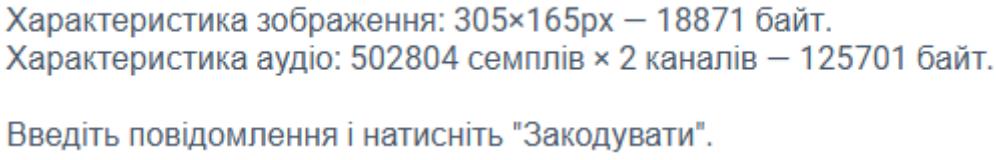
Після чого проведено приховування. Повідомлення приховано з наступним розподілом інформації (рис. 3.4).



Зображення 3.4 - результат тестування динамічного розподілу інформації програмно реалізованого вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

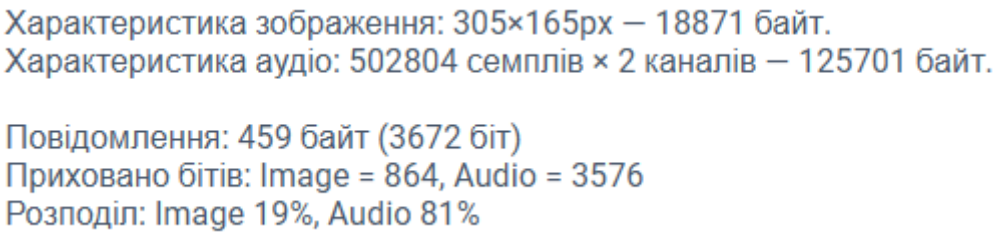
Результати тестування наступні: в фото носій приховано 72% інформації, в аудіо файл приховано 28% інформації.

Після чого проведено другий етап тестування де обрано те ж саме повідомлення для приховування об`ємом 459 символи, аудіо носій тривалістю 00:10 хв та фото носій розміром 374 на 377 пікселів, який значно меншого розміру ніж під час попереднього тестування (рис. 3.5).



Зображення 3.5 - характеристика обраних носіїв (зображення та аудіо файлу).

Після чого проведено успішне приховування інформації в обрані носії з наступним розподілом (рис. 3.6).



Зображення 3.6 - результат тестування динамічного розподілу інформації програмно реалізованого вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Результати тестування наступні: в фото носій приховано 19% інформації, в аудіо файл приховано 81% інформації.

В результаті проведеного тестування можна дійти висновку що алгоритм дійсно ефективний, адже розділяє навантаження для приховування між носіями на основі метрик самих носіїв.

Після цього було проведено тестування цілісності і достовірності повідомлення після його приховування.

Для цього тестування було обрано випадковий фрагмент тексту для проведення приховування (рис. 3.7).

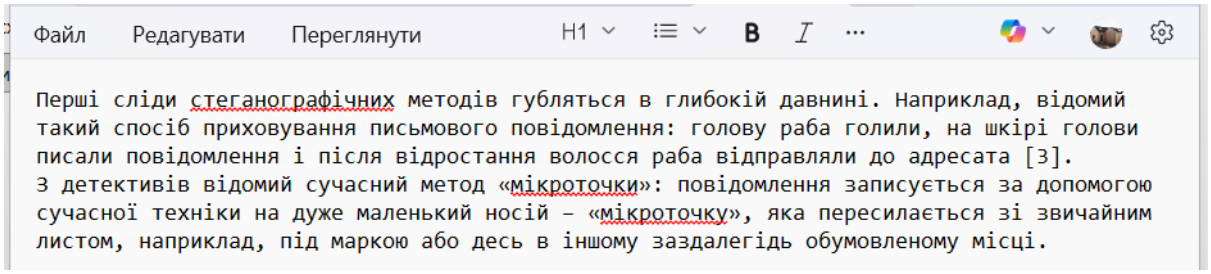
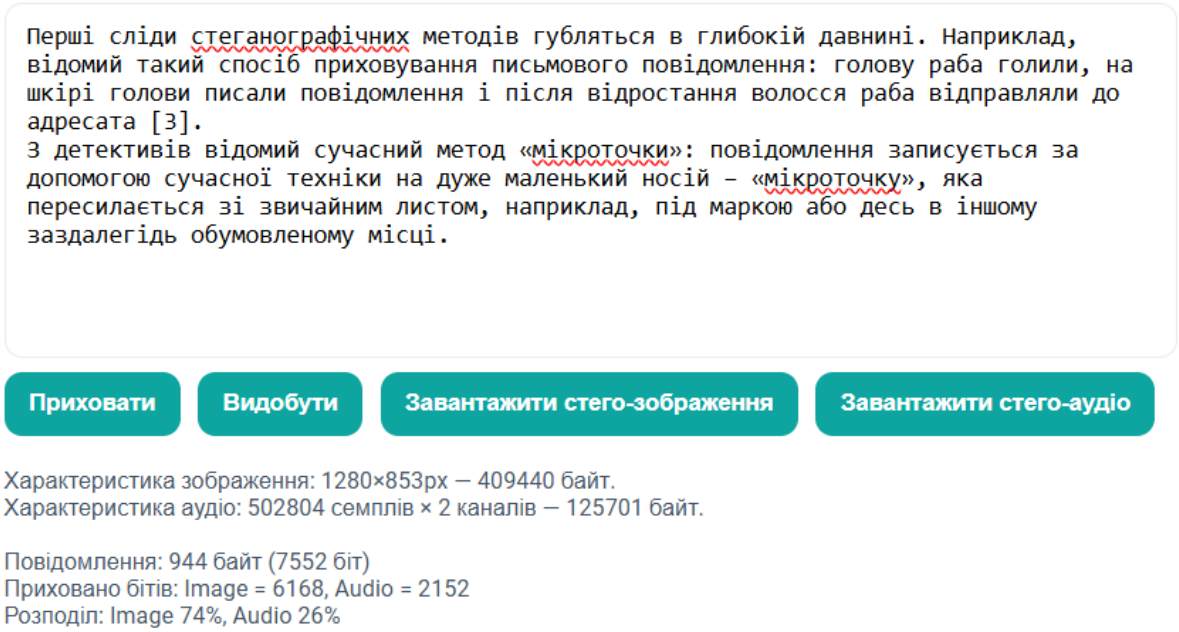


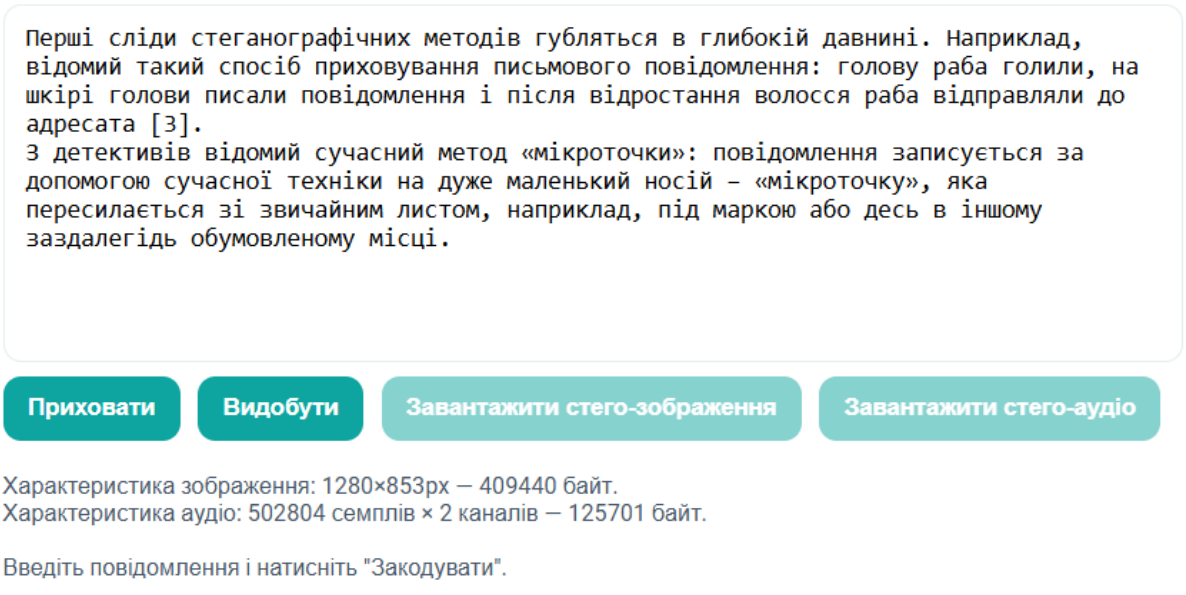
Рисунок 3.7 – тест який обрано для приховування.

Після чого проведено його приховування в 2 носії (фото та аудіо файл) (рис. 3.8).



Зображення 3.8 - результат динамічного розподілу приховуваної інформації між 2-ма носіями.

Після чого збережено сформовані стего-файли і проведено видобуток в зворотньому напрямку, отримано наступне повідомлення (рис. 3.9).



Зображення 3.9 - результат тестування цілісності і достовірності інформації після її видобутку в програмно реалізованому вдосконаленому багатоканальному стеганографічному методі з використанням динамічного розподілу прихованої інформації.

В результаті даного тестування можна дійти висновку що повідомлення повністю ідентичне і нічим не відрізняється від початкового, значить метод повністю працездатний і виконує свою основну функцію.

Наступним було проведено тестування якості носіїв після приховування в них інформації. Це тестування пропонується провести в 2 етапи.

На першому етапі приховано інформацію в зображення та порівняно початкове зображення з тим в яке приховано інформацію (рис. 3.10; 3.11).

Рисунок 3.10 – початкове зображення.

Рисунок 3.11 – зображення з прихованою інформацією.

Також порівняно властивості цих 2-х файлів (рис. 3.12; 3.13).

Розташування:	C:\Users\vital\Downloads
Розмір:	472 КБ (484 059 байтів)
На диску:	480 КБ (491 520 байтів)

Рисунок 3.12 – властивості початкового зображення.

Розташування:	C:\Users\vital\Downloads
Розмір:	2,59 МБ (2 725 532 байтів)
На диску:	2,60 МБ (2 727 936 байтів)

Рисунок 3.13 – властивості зображення в яке приховано інформацію.

Зображення в яке приховано інформаціє більше за розміром ніж початкове, але це нормально для методу комп’ютерної стеганографії LSB.

На другому етапі тестування приховано інформацію в аудіо файл та порівняно початковий файл з тим в яке приховано інформацію. Аналіз було виконано за допомогою онлайн ресурсу Editor.Audio (рис. 3.14; рис. 3.15) [52].

Рисунок 3.14 – доріжка початкового аудіо файлу.

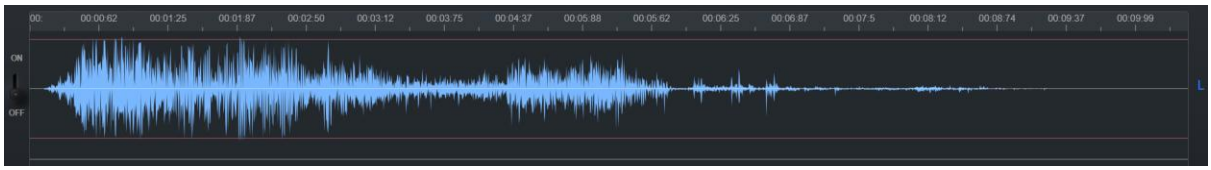


Рисунок 3.15 – доріжка аудіо файлу в яке приховано інформацію.

Також порівняно властивості цих 2-х файлів (рис. 3.14; 3.15).

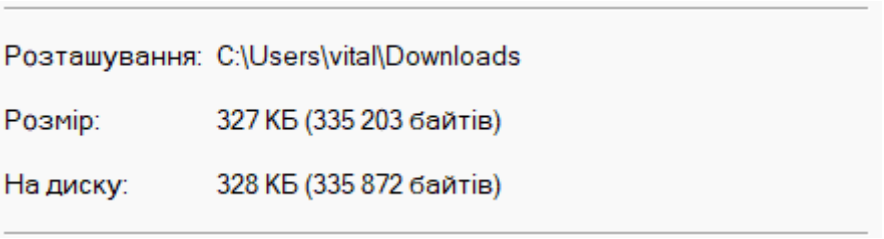


Рисунок 3.14 – властивості початкового аудіо файлу.

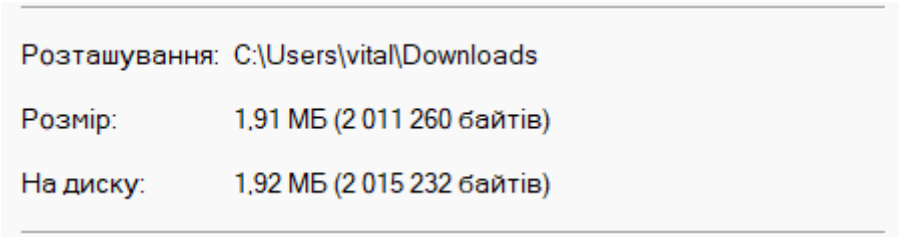


Рисунок 3.15 – властивості аудіо файлу в яке приховано інформацію.

Вихідний аудіо файл з прихованою інформацією також більший за розміром за початковий файл, це норма для методу LSB.

Після цього пропоную провести тестування криптографічної стійкості методу. Для цього було приховано повідомлення в 2 носії і

здійснено спробу його несанкціонованого отримання не вказавши пароль (рис. 3.16).

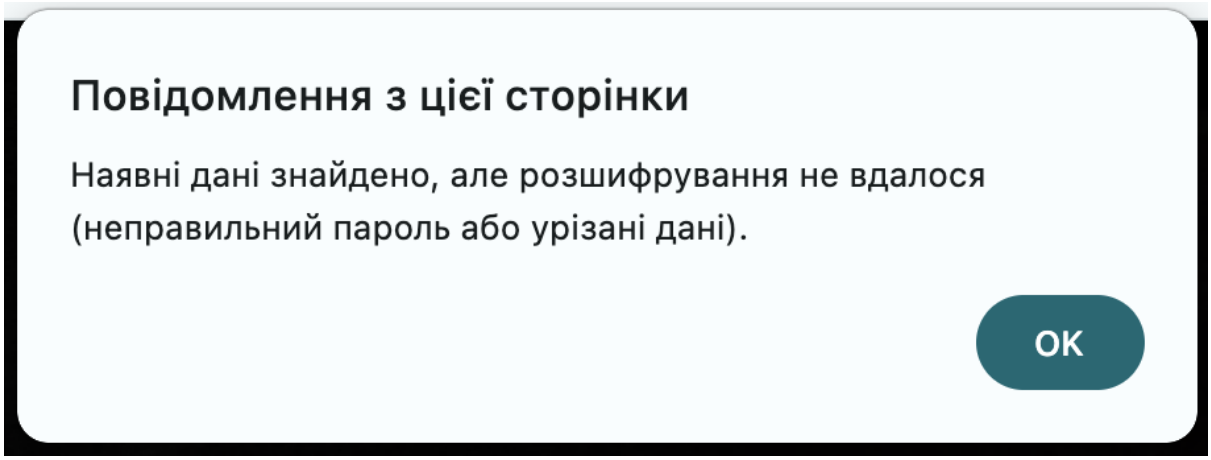


Рисунок 3.16 – результат після не вірно введеного пароля.

Також було здійснено спробу видобутку інформації з стего-файлів вказавши не вірний пароль (рис. 3.17).

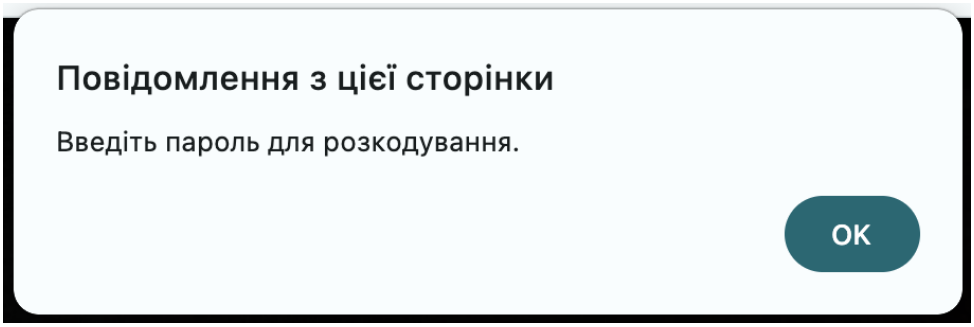


Рисунок 3.17 – результат після не введеного пароля

Також проведено тестування можливості одноканального приховування інформації, за умови якщо один із каналів відсутній (не обраний). Для цього було здійснено спробу приховування інформації вибравши лише 1 із пропонованих каналів.

Виконано спробу приховування лише в канал зображення (рис. 3.18).

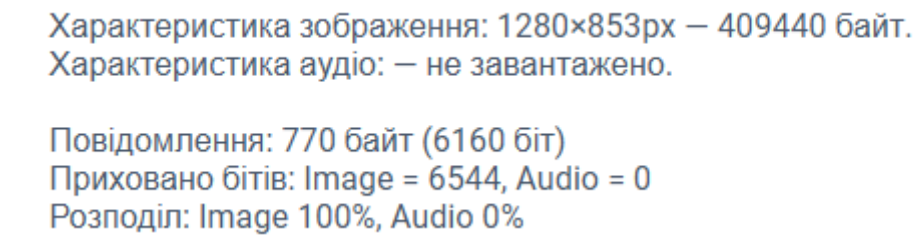


Рисунок 3.18 – результат розподілу повідомлення між каналами.

При цьому розподіл повідомлення відбувся наступним чином 100% в зображення, 0% в аудіо (через відсутність каналу).

Приховування лише в канал аудіо файлу (рис. 3.19).

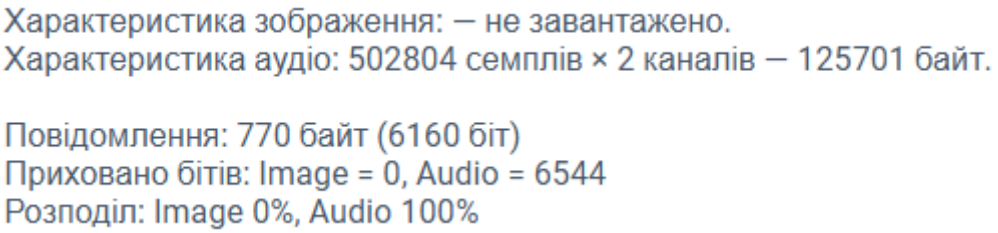


Рисунок 3.19 – результат розподілу повідомлення між каналами.

При цьому розподіл повідомлення відбувся наступним чином 0% в зображення (через відсутність каналу), 100% в аудіо.

Виходячи з результатів проведених тестувань, можна зробити висновок що вдосконалений багатоканальний стеганографічний метод з використанням динамічного розподілу прихованої інформації, є дійсно вдосконаленим і ефективнішим, кращим в порівнянні з існуючими версіями багатоканальних стеганографічних методів, це підтверджують результати тестувань.

3.4. Висновки до розділу 3.

В даному розділі було обґрунтовано вибір мови програмування та середовища розробки, окреслено недоліки та ключові переваги обраних рішень. Також було здійснено програмну реалізацію вдосконаленого

багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації, детально розписано процес розробки.

Також проведено тестування програмно реалізовано вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації. Проведено наступні тестування:

- Тестування на продуктивність та масштабованість;
- Тестування динамічного розподілу;
- Тестування цілісності повідомлення після приховування та отримання у зворотньому порядку;
- Оцінка якості носіїв після вбудовування;
- Тестування криптографічної стійкості;
- Можливість одноканального приховування.

У результаті опрацювання даного розділу дипломної роботи стало очевидно, що вдосконалений багатоканальний стеганографічний метод із використанням динамічного розподілу прихованої інформації демонструє справді вищу ефективність порівняно з існуючими підходами. Поглиблений аналіз та узагальнення експериментальних даних засвідчили, що запропонована технологія забезпечує не лише підвищений рівень стійкості до виявлення, а й оптимальні характеристики щодо збереження якості контейнерів, що є критично важливим у контексті сучасних вимог до інформаційної безпеки. Результати тестувань підтвердили, що гнучкість динамічного розподілу навантаження та здатність методу адаптивно реагувати на параметри каналів дозволяють суттєво розширити можливості багатоканальної стеганографії, роблячи цей підхід більш надійним, продуктивним і практично орієнтованим.

4. ЕКОНОМІЧНА ЧАСТИНА РОЗРОБКИ ВДОСКОНАЛЕНОГО БАГАТОКАНАЛЬНОГО СТЕГАНОГРАФІЧНОГО МЕТОДУ З ДИНАМІЧНИМ РОЗПОДІЛОМ ІНФОРМАЦІЇ

Науково-дослідні та інноваційні розробки у будь якій сфері та в сфері кібербезпеки особливо мають високий потенціал практичного застосування і розвитку виключно за умови їх ефективності на практиці та економічної доцільності. В умовах постійного та безперервного активного розвитку цифрових технологій особливо актуальними стають методи забезпечення конфіденційності та цілісності інформації, які можуть поєднувати високу стійкість до їх виявлення з оптимальними витратами на впровадження. Економічна оцінка цих розробок дає змогу визначити їхню конкурентоспроможність на ринку і доцільність в подальшій комерційній реалізації та використанні.

Магістерська кваліфікаційна робота на тему: «Вдосконалення багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації» відноситься до науково-технічної розробки, яка спрямована на підвищення рівня захисту даних під час передавання інформації у відкритих мережах та відкритими каналами зв’язку. Запропонований вдосконалений стеганографічний метод має комерційний потенціал для впровадження у практичних системах інформаційної безпеки, зокрема у сфері захисту інтелектуальної власності, безпечного обміну корпоративними даними та комунікацій в мережі Інтернет.

Метою даного розділу є оцінка економічної ефективності та доцільності впровадження вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації, аналіз його комерційного потенціалу, визначення витрат на розробку і реалізацію повноцінного програмного забезпечення, а також обґрунтування

доцільності подальшого використання результатів роботи у практичній діяльності.

4.1. Оцінювання комерційного потенціалу розробки.

Метою комерційного та технологічного аудиту є визначення науково-технічного рівня та оцінка комерційного потенціалу вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації, розробленого в рамках магістерської кваліфікаційної роботи. Аудит спрямований на аналіз інноваційності та ринкової привабливості створеної технології для її подальшої комерціалізації і реальному застосуванні [53].

Для проведення технологічного аудиту було розроблено таблицю 4.1 в якій здійснено оцінку комерційного потенціалу вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації за 12-ма критеріями та за 5-ти бальною школою оцінювання (від 1 до 5-ти) [54].

Таблиця 4.1. Критерії оцінки комерційного потенціалу вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Критерій	Критерії оцінювання та бали від 1 до 5-ти				
	1	2	3	4	5
Технічна реалізація концепції					
1	Концепція не підтверджена практично	Є лише теоретичне моделювання	Концепція підтверджена попередніми експерементами	Концупція перевірена на тестованих даних	Перевірено працездатність в реальних умовах на практиці
2	Реалізація неможлива сучасними засобами	Потрібне значне доопрацювання технологій	Реалізація потребує спеціалізованого середовища	Реалізація можлива стандартними інструментами	Реалізація здійснюється з використанням відкритих і доступних технологій

Продовження таблиці 4.1

Ринкові переваги та недоліки					
3	Багато існуючих аналогів на ринку із кращими показниками	Є кілька аналогів зі схожими характеристиками	Є аналоги але з гіршими показниками	Існує лише один конкурентний аналог	Аналоги відсутні
4	Розробка значно дорожча за аналоги	Розробка трохи дорожча за аналоги	Ціна на рівні аналогів	Ціна розробки трохи нижча за аналоги	Ціна розробки значно нижча за аналоги
5	Ефективні результати гірші за існуючі методи	Ефективність на рівні існуючих аналогів	Трохи краща ефективність	Значно краща ефективність у типових задачах	Висока ефективність у всіх тестових сценаріях
Ринкові перспективи					
6	Малий потенціал без перспектив росту	Малий, але зростаючий ринок	Середній ринок зі стабільним попитом	Великий стабільний ринок	Великий ринок з високим потенціалом росту
7	Висока конкуренція провідних компаній	Висока конкуренція серед малих компаній	Помірна конкуренція	Незначна конкуренція	Відсутність конкурентів на ринку у цій ніші
Практична реалізація					
8	Відсутні фахівці з даної галузі	Потрібне додаткове навчання персоналу	Частково є підготовлені фахівці	Є кваліфіковані спеціалісти у команді	Повна готовність фахівців до реалізації
9	Відсутні кошти та джерела фінансування на розробку	Є потенційні джерела, але без гарантій	Джерела фінансування частково визначені	Є забезпечене фінансування	Не потребує значних фінансових витрат
10	Відсутня технічна база	Потребує закупівлі дорогого обладнання	Потребує незначного оновлення обладнання	Використовується стандартна апаратура	Використовується наявна техніка без додаткових витрат

Продовження таблиці 4.1

11	Тривалість впровадження понад 10 років	Тривалість впровадження від 5-ти до 10-ти років	Тривалість впровадження від 3-х до 5-ти років	Тривалість впровадження від 1-го до 3-х років	Тривалість впровадження менше 1-го року
12	Потребує спеціальних ліцензій та сертифікацій	Потребує тривалої сертифікації	Потрібна мінімальна реєстрація	Потрібне лише повідомлення про впровадження	Відсутні будь-які регуляторні обмеження

Наступним розроблено таблицю 4.2. з результатами оцінки критеріїв комерційного потенціалу вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації [55].

Таблиця 4.2. – результататати оцінки критеріїв комерційного потенціалу вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Середньоарифметична сума балів, розрахована на основі висновків визначених оцінок	Рівень комерційного потенціалу розробки
12-20	Низький
21-32	Нижче середнього
33-44	Середній
45-54	Вище середнього
55-60	Високий

Результати оцінок 3-х незалежних експерти (Павленко Максим Сергійович – керівник проєктів із впровадження систем безпеки; Литвиненко Ірина Вікторівна – старший аналітик з кіберзахисту; Ковальчук Дмитро Олександрович – консультант із цифрової безпеки та криптографії) критеріїв комерційного потенціалу вдосконаленого стеганографічного

методу з використанням динамічного розподілу прихованої інформації наведено в таблиці 4.3.

Таблиця 4.3 – результати з оцінками критеріїв комерційного потенціалу вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Критерій	Незалежна експертна оцінка			
	Павленко Максим Сергійович	Литвиненко Ірина Вікторівна	Ковальчук Дмитро Олександрович	Середня оцінка
1 (Підтвердження достовірності концепції)	5	5	4	14
2 (Рівень технічної складності реалізації)	5	4	4	13
3 (Наявність аналогів)	4	5	5	14
4 (Цінова політика реалізації)	4	4	5	13
5 (Ефективність вдосконаленого методу)	4	4	4	12
6 (Розмір та динаміка ринку)	2	5	4	11
7 (Конкуренція на ринку)	3	4	4	11
8 (Кадрове забезпечення)	3	4	4	11
9 (Фінансові ресурси)	1	5	5	11
10 (Матеріально-технічна база)	4	5	4	13
11 (Тривалість впровадження)	5	4	5	14
12 (Наявність регуляторних та нормативно-правових бар'єрів)	4	5	5	14
Сума балів	$СБ_1 = 44$	$СБ_2 = 54$	$СБ_3 = 53$	$СБ_{заг} = 151/180$
$СБ = \frac{\sum_3^1 СБ_i}{3} = \frac{44 + 54 + 53}{3} = 50{,}3$				

Загальна середньостатистична оцінка критеріїв комерційного потенціалу вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації становить 50,3 балів, що згідно таблиці 4.2 є «Вище середнього».

Це означає, що вдосконалений багатоканальний стеганографічний метод із використанням динамічного розподілу прихованої інформації має високі перспективи подальшого розвитку, впровадження, використання та комерціалізації.

4.2. Прогнозування витрат на виконання науково-дослідної роботи.

Під час розрахунку собівартості розробки, враховуються витрати, пов’язані з проведенням науково-дослідної роботи та групуються за такими статтями [56]:

- основна заробітна плата виконавців робіт;
- додаткова заробітна плата;
- матеріальні витрати (податки);
- витрати на спеціальне обладнання для наукових (експериментальних) робіт;
- амортизаційні відрахування;
- витрати на електроенергію, воду, паливо та інші енергоресурси;
- витрати на послуги сторонніх організацій
- витрати на службові відрядження;
- програмне забезпечення для наукових робіт;
-інші витрати.

До інших витрат під час планування, обліку та калькулювання собівартості робіт можуть належати витрати, які не включаються безпосередньо до основних статей калькуляції, але є необхідними для

виконання робіт та забезпечення їх належного рівня якості в кінцевому результаті.

До таких витрат можуть відноситись оплата відряджень працівників які пов’язані із виконанням робіт, участю їх у наукових конференціях, семінарах, консультаціях з фахівцями інших організацій в галузі розробки чи у сфері права, а також витрати на підготовку технічної документації, переклади, оформлення звітів чи заявок на патенти.

Також до цієї категорії належать витрати на забезпечення та проведення навчань персоналу щодо охорони праці, техніки безпеки, пожежної безпеки, а також витрати, пов’язані з утилізацією відходів, що виникають у процесі виконання робіт над проектом.

До інших витрат також відносяться витрати на амортизацію малоцінних і швидкозношуваних предметів (розхідників), що використовуються під час проведення експериментів або випробувань, витрати на ремонт обладнання.

Крім того, до інших витрат можуть бути включені податки, різного роду збори та інші обов’язкові платежі пов’язані безпосередньо з виконанням робіт. Також сюди відносяться непередбачувані витрати, що виникають у процесі досліджень (ті які не можливо передбачити чи спрогнозувати), якщо вони документально підтверджені та обґрунтовані.

1. Розрахунок основної заробітної плати працівників розраховується за формулою [57]:

$$З_о = \sum_{i=1}^{k} \frac{М_{по} * t_і}{T_r} \tag{4.1}$$

де, k – кількість залучених працівників;

$М_{по}$ – місячний посадовий оклад конкретного працівника;

$t_і$ – кількість робочих днів над проектом конкретного працівника;

T_r – середня кількість робочих днів в місяці.

Для розробки потрібно залучити керівника проекту з посадовим окладом 33 000 грн на місяць, при середні кількості робочих днів в місяць 22 дні, за загальні потребі робочих днів 36 днів.

Таким чином згідно формули розраховано витрати на заробітну плату керівника проекту (Project Manager):

$$З_о = \frac{33000 * 36}{22} = 54000 \text{ грн.}$$

Також для розробки вдосконаленого стеганографічного методу з динамічним розподілом інформації, необхідно залучити штатного розробника ПЗ з посадовим окладом 27 500 грн на місяць, при середні кількості робочих днів в місяць 22 дні, за загальні потребі робочих днів 41 день.

По ті ж самі формулі розраховано витрати на заробітну плату штатному розробнику ПЗ:

$$З_о = \frac{27500 * 41}{22} = 51250 \text{ грн.}$$

Таблиця 4.4 – Витрати на заробітну плату розробників.

Назва посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Кількість робочих днів над проектом	Загальні витрати на основну заробітну плату, грн
Project Manager	33 000	1500	36	54 000
Штатний розробник ПЗ	27 500	1250	41	51 250
Науковий керівник	16 060	730	5	3 650
Всього:				108 900,00 грн

Згідно розрахунків на розробку вдосконаленого стеганографічного методу з динамічним розподілом інформації, необхідно залучити 3

працівники (Project Manager, штатного розробника ПЗ та наукового керівника), та витратити на їх основну заробітну плату 108 900,00 грн.

2. Додаткова заробітна плата розраховується як додаткових 12% від основної заробітної плати кожного працівника, та розраховується за формулою:

$$З_д = (З_о + З_р) * \frac{Н_{дод}}{100\%} \quad (4.2.)$$

Таким чином додаткова заробітна плата для працівників проекту становить:

$$З_д = 54\,000 \cdot 0.12 = 6\,480 \text{ грн}$$

Та додаткова заробітна плата для штатного розробника ПЗ становить 6 150 грн.:

$$З_д = 108\,900 \cdot 0.12 = 13\,068 \text{ грн}$$

3. Відрахування на соціальні заходи.

До статті «Відрахування на соціальні заходи» належать відрахування внеску на загальнообов'язкове державне соціальне страхування та для здійснення заходів щодо соціального захисту населення (ЄСВ – єдиний соціальний внесок).

Нарахування на заробітну плату дослідників та робітників розраховується як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

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

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

Відрахування на соціальні заходи з project manager:

$$З_н = (54\,000 + 6\,480) * \frac{22}{100\%} = 13\,305{,}6$$

Відрахування на соціальні заходи з штатного розробника ПЗ:

$$З_н = (51\,250 + 6\,150) * \frac{22}{100\%} = 12\,628$$

Відрахування на соціальні заходи з наукового керівника:

$$З_н = (3\,650 + 438) * \frac{22}{100\%} = 899{,}36$$

Загальні витрати з статті «Відрахування на соціальні заходи» на всіх залучених працівників становлять:

$$13305{,}6 + 12\,628 + 899{,}36 = 26\,832{,}96$$

4. Товаро-матеріальні цінності.

До статті «Товаро-матеріальні цінності» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби чи предмети праці, які були придбані у сторонніх підприємств, установ або організацій і використані безпосередньо для виконання досліджень відповідно до їхнього прямого призначення та встановлених нормативів витрачання. До цієї категорії також включаються витрачені напівфабрикати, що потребують монтажу, доопрацювання або виготовлення всередині організації, якщо вони є необхідними для реалізації науково-дослідних робіт.

Окрім цього, стаття охоплює витрати на дослідні зразки, які були виготовлені сторонніми виробниками на основі технічної документації, що розроблена науковою організацією. Подібні витрати формують матеріальну основу процесу дослідження, оскільки саме через них забезпечується можливість створення експериментальних моделей, проведення вимірювань, тестувань та перевірки гіпотез. Таким чином, стаття «Товаро-матеріальні цінності» відображає ключові елементи ресурсного забезпечення наукових робіт і є важливою складовою загальної структури витрат на проведення дослідницької діяльності.[58].

Витрати на матеріали (М) у вартісному вираженні розраховуються окремо для кожного виду матеріалів за формулою:

$$M = \sum_{j-1}^{n} H_j * Ц_j * К_j - \sum_{j-1}^{n} В_j * Ц_{вj}$$

(4.4.)

Спершу визначено товаро-матеріальні цінності ТМЦ які були використані як розхідник під час виконання робіт.

Таблиця 4.5 – використані ТМЦ під час виконання робіт.

Найменування ТМЦ	Ціна за одиницю, грн.	Використано	Загальна вартість використаних ТМЦ
Папір	190	1	190
Канцелярські вироби	80	3	240
USB накопичувач	260	1	260
Разом:			690 грн.

Таким чином витрати на матеріали (М) з урахуванням вартості транспортування розраховуються окремо для кожного виду матеріалів та становлять:

$$В_м = 690 * 2\% = 703{,}8.$$

5. Розрахунок витрат на комплектуючі:

Витрати на комплектуючі вироби (Кв), які використовують при дослідженні нового технічного рішення, розраховуються, згідно з їхньою номенклатурою, за формулою:

$$К_в = \sum_{j-1}^{n} Н_j * Ц_j * К_j$$

(4.5)

де $Н_j$ – кількість комплектуючих j-го виду, шт.;

$Ц_j$ – покупна ціна комплектуючих j-го виду, грн;

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

Витрати на комплектуючі для розробки занесено в таблицю 4.7.

Таблиця 4.7. – витрати на комплектуючі для розробки.

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Зовнішній SSD 1 ТБ	1	3000	3000
USB 3.0 флеш накопичувач 128 ГБ	1	400	400
Raspberry Pi 4 (для тестових стендів)	1	2500	2500
Охолоджуючий вентилятор / кулер	1	200	200
Ethernet / USB-кабель	1	150	150
Разом:			6 250 грн.

$К_j$ = 1,10 — стандартні 10%

Для кожної позиції обчислюємо: $Н_j * Ц_j * К_j$

Зовнішній SSD: 1 · 3 000 · 1,10 = 3 300 грн

USB флеш: 1 · 400 · 1,10 = 440 грн

Raspberry Pi 4: 1 · 2 500 · 1,10 = 2 750 грн

Вентилятор: 1 · 200 · 1,10 = 220 грн

Кабель: 1 · 150 · 1,10 = 165 грн

Загальні витрати на комплектуючі з урахуванням вартості доставки становлять:

3 300 + 440 + 2 750 + 220 + 165 = 6 875 грн.

6. Для розрахунку амортизаційних відрахувань необхідно враховувати вартість ТМЦ яке використовувалось під час розробки у співвідношенні до заявленого виробником терміну служби. Річна амортизація розраховується за наступною формулою [59]:

$$А_{обл} = \frac{Ц_б}{Т_в} * \frac{t_{вих}}{12} \quad (4.6.)$$

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

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

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

Таким чином під час виконання магістерської кваліфікаційної роботи використовувався власний ПК вартістю 45 000 грн., ліквідаційна вартість якого становить 15 000, а заявлений виробником термін експлуатації становить 2 роки.

$$А = \frac{45\,000}{2} * \frac{3}{12} = 22\,500 \cdot 0{,}25 = 5\,625 \text{ грн}$$

Виходячи з розрахунків амортизаційні витрати на техніку склали 3 750 грн.

7. Для програмної реалізації використовувалось середовище розробки Visual Studio Code (VS Code) яке є безкоштовним. Тому витрат на програмне забезпечення не було.

8. Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуються як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$І_в = (З_о + З_р) * \frac{Н_{ів}}{100\%} \tag{4.7}$$

Таким чином для 50% інші витрати становлять:

$$І_в = 108\,900 * 0{,}50 = 54\,450$$

Для 75%:

$$І_в = 108\,900 * 0{,}75 = 81\,675$$

Для 100%:

$$I_в = 108\,900 * 1 = 108\,900$$

Для підрахунку всіх прогнозованих витрат додано всі раніше проведені розрахунки в таблицю 4.8.

Таблиця 4.8 – загальний підсумок всіх витрат для розробки і впровадження вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Стаття витрат	Сума витрат (грн.)
Основна зарплата	108 900
Додаткова зарплата	13 068
ЄСВ	26 832,96
Матеріали	703.80
Комплектуючі	6 875
Амортизація	5 625
Разом	162 004,76 грн

Таким чином загальний прогнозований бюджет розробки становить: 162 004,76 грн.

Прогнозування загальних втрат ЗВ на виконання та впровадження результатів виконаної МКНР здійснюється за формулою:

$$ЗВ = \frac{В}{\eta},$$

(4.8)

де η – коефіцієнт, який характеризує стадію виконання даної НДР.

Так як розробка знаходиться на етапі розробки технології, то η = 0.4

$$ЗВ = \frac{162\,004,76}{0.4} = 405\,011,9$$

4.3. Розрахунок економічної ефективності науково-технічної розробки.

У сучасних ринкових умовах головним критерієм ефективності науково-технічних інновацій є рівень економічної вигоди, який може отримати інвестор або підприємство після впровадження даної розробки.

В межах даної магістерської кваліфікаційної роботи здійснено прогнозування комерційних та економічних результатів та матеріальних вигод від реалізації та впровадження вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Очікується, що комерціалізація результатів розробки відбуватиметься протягом трьох років з моменту розробки, упродовж яких активно прогнозується зріст кількості користувачів розробленого продукту завдяки підвищенню його захищеності, ефективності та надійності [60].

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

$$\Delta П_i = \sum_1^n (\Delta D_0 * N * D_0 * \Delta N)_i * \lambda * p * \left(1 - \frac{u}{100}\right) \quad (4.9)$$

де, ΔD_0 – покращення основного оціночного показника від впровадження результатів розробки у даному році;

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

ΔN - покращення основного кількісного показника діяльності підприємства після впровадження результатів розробки:

D_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадженої розробки;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт λ = 0,83 (20%);

ρ – коефіцієнт, який враховує рентабельність продукту: ρ = 0,2 (20%);

υ – ставка податку на прибуток – 18 (станом на 2025 рік – 18%).

Розрахунок економічного ефекту згідно формули на 1-й рік після впровадження:

$$\Delta П_i = (5000 * 350 + 85000 * 50) * 0{,}83 * 0{,}2 * \left(1 - \frac{0{,}18}{100}\right) = 994\,213 \text{ грн.}$$

Розрахунок економічного ефекту згідно формули на 2-й рік після впровадження:

$$\Delta П_i = (5000 * 350 + 85000 * (50 + 70)) * 0{,}83 * 0{,}2 * \left(1 - \frac{0{,}18}{100}\right) = 1\,982\,430 \text{ грн.}$$

Розрахунок економічного ефекту згідно формули на 3-й рік після впровадження:

$$\Delta П_i = (5000 * 350 + 85000 * (50 + 70 + 100)) * 0{,}83 * 0{,}2 * \left(1 - \frac{0{,}18}{100}\right) = 3\,387\,470 \text{ грн.}$$

Загальний прогнозований економічний ефект за 3 роки:

$$\Delta П_{заг} = 994\,213 + 1\,982\,430 + 3\,387\,470 = 6\,364\,113 \text{грн}$$

Таким чином було проведено розрахунок економічної ефективності науково-технічної розробки на 3 роки.

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1-\gamma)^t} \quad (4.12)$$

Таким чином приведена вартість збільшення всіх чистих прибутків ПП становить:

$$ПП = \frac{994\ 213}{1{,}12} + \frac{1\ 982\ 430}{1{,}12^2} + \frac{3\ 387\ 470}{1{,}12^3}$$

$$= 887\ 688 + 1\ 579\ 700 + 2\ 409\ 260 = 4\ 876\ 648 \text{ грн}$$

Необхідно здійснити розрахунок внутрішньої економічної дохідності в показнику Е, тобто визначити внутрішню норму дохідності інвестицій (IRR, Internal Rate of Return), яка відображає такий рівень рентабельності проєкту, за якого чистий приведений дохід дорівнює нулю. Внутрішня норма дохідності є одним із найточніших критеріїв інвестиційної привабливості, оскільки вона показує максимальну ставку дисконту, за якої інвестиційний проєкт залишається економічно виправданим. Фактично це граничне значення ефективності, що характеризує реальну здатність розробки генерувати дохід у майбутньому.

Після отримання значення IRR його необхідно порівняти з бар'єрною ставкою дисконтування, яка визначає мінімально прийнятний рівень внутрішньої дохідності для інвестора або організації. Якщо розрахована внутрішня норма дохідності перевищує бар'єрну ставку, інвестиції в науково-технічну розробку є економічно обґрунтованими та потенційно вигідними. Якщо ж IRR виявляється нижчим за цей пороговий показник, вкладати кошти у відповідний проєкт недоцільно, оскільки він не забезпечить достатнього рівня прибутковості. Таким чином, порівняння внутрішньої норми дохідності з бар'єрною ставкою дозволяє об'єктивно

оцінити доцільність інвестування та перспективність науково-технічної розробки у довгостроковій фінансовій перспективі.

Далі розраховують величину початкових інвестицій PV , які потенцій ний інвестор має вкласти для впровадження і комерціалізації науково технічної розробки. Для цього можна використати формулу:

$$PV = K_{\text{інв}} * \text{ЗВ} \tag{4.13}$$

де $K_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науко во-технічної розробки та її комерціалізацію. Це можуть бути витрати на під готовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай інв k =2...5, але може бути і більшим;

В даному випадку $K_{\text{інв}} = 2$

$$PV = 2 * 405\,011{,}9 = 810\,023{,}8$$

Тоді абсолютний економічний ефект абс Е або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впрова дження та комерціалізації науково-технічної розробки становитиме:

$$\text{Е}_{\text{абс}} = \text{ПП} - PV \tag{4.14}$$

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

$$\text{Е}_{\text{абс}} = 4\,876\,648 - 810\,023{,}8 = 4\,066\,624{,}2$$

Внутрішня економічна дохідність інвестицій в Е , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково технічної розробки, розраховується за формулою:

$$\text{Е}_{\text{в}} = \sqrt[\text{т}]{1 + \frac{\text{Е}_{\text{абс}}}{PV}} - 1 \tag{4.15}$$

У даному випадку інвестиції умовно приймаються як початкові витрати на розробку (PV)

$$\text{Е}_{\text{абс}} = \ 4\ 066\ 624{,}2 \text{ грн}$$

$$T = 3 \text{ роки}$$

$$\text{Е}_{\text{в}} = \sqrt[3]{1 + \frac{4\ 066\ 624{,}2}{810\ 023{,}8}} - 1 \approx 1{,}71 \ \approx 171\%$$

Таким чином внутрішня норма дохідності на 3 роки становить 171%.

Далі необхідно визначити бар'єрну ставку дисконтування $t_{\text{мін}}$, тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Мінімальна внутрішня економічна дохідність вкладених інвестицій мін визначається за формулою:

$$t_{\text{мін}} = b + f \tag{4.16}$$

Де b – депозитна ставка (0,10),

f – ризик (0,20).

$$t_{\text{мін}} = \ 0{,}10 + 0{,}20 = 0{,}30 = 30\%$$

Так як $\text{Е}_{\text{в}} > t_{\text{мін}}$ то інвестор може бути зацікавлений у фінансуванні даної наукової розробки

4.4. Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Оцінювання економічної ефективності впровадження науково-технічної розробки передбачає комплексний аналіз співвідношення між обсягом інвестицій, необхідних для реалізації проєкту, та величиною економічного ефекту, який досягається в результаті його практичного використання. Такий підхід дає змогу об'єктивно визначити, наскільки

раціональним є вкладення коштів у запропоновану технологію, а також оцінити її здатність забезпечувати додану вартість у довгостроковій перспективі. Процес оцінювання охоплює розрахунок очікуваних витрат, прогнозування можливих доходів, а також визначення строків, протягом яких ці доходи зможуть компенсувати початкові інвестиції.

Ключовими показниками, що дозволяють визначити рівень ефективності інвестицій, є коефіцієнт економічної ефективності (Е) та період окупності інвестицій (Т_ок). Перший показник відображає відношення річного економічного ефекту до загального обсягу інвестицій і характеризує прибутковість проєкту. Другий визначає час, необхідний для повного повернення вкладених коштів, і служить індикатором інвестиційних ризиків та швидкості віддачі. Разом ці показники формують цілісну картину економічної доцільності впровадження науково-технічної розробки та дозволяють оцінити перспективи її широкого використання і подальшої комерціалізації [61].

Отримане значення $E = 1{,}71$ свідчить про те, що на кожну вкладену гривню інвестицій очікується економічна віддача у розмірі 1,71 грн, тобто ефективність вкладень перевищує 100%, що є позитивним показником для науково-технічних проєктів інноваційного спрямування.

Період окупності інвестицій визначається за формулою:

$$\mathrm{T_{ок}} = \frac{\mathrm{I}}{\mathrm{E_в}} \tag{4.17}$$

Підставивши відповідні значення, отримаємо:

$$\mathrm{T_{ок}} = \frac{1}{1{,}71} = 0{,}585 \text{ року}$$

$$0{,}585 < 3 \text{ років}$$

$$0{,}585 * 12 \approx 7{,}02 \text{ місяців}$$

Таким чином, період окупності інвестицій становить приблизно 7 місяців, що свідчить про високу ефективність та швидку віддачу вкладених коштів.

Показники ефективності підтверджують економічну доцільність впровадження вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації у практичних системах захисту даних. Отримані результати демонструють, що інвестиції в дану технологію можуть бути повністю окуплені менш ніж за рік, після чого розробка приноситиме стабільний прибуток у подальші роки експлуатації та використання.

Отже науково-технічна розробка вважається економічно ефективною та інвестиційно привабливою.

4.5. Висновки до розділу 4.

В даному розділі було проведено оцінку комерційного потенціалу вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації. Оцінку було проведено за 12-ма пунктами по 5-ти бальні шкалі, в результаті сформовано відповідні висновки.

Також в ході виконання даного розділу було проведено розрахунок собівартості розробки з урахуванням витрат пов’язаних з проведенням науково-дослідної роботи які групуються за наступними статтями:

- основна заробітна плата виконавців робіт;
- додаткова заробітна плата;
- матеріальні витрати (податки);
- витрати на спеціальне обладнання для наукових (експериментальних) робіт;
- амортизаційні відрахування;
- програмне забезпечення для наукових робіт;

-інші витрати.

В результаті було отримано загальний прогнозований бюджет розробки вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Після чого було розраховано коефіцієнт економічної ефективності та період окупності вкладених інвестицій інвестором.

ВИСНОВКИ

В ході виконання даної дипломної роботи було проаналізовано основні засади поняття стеганографії та її роль у забезпечення інформаційної безпеки. Проведено огляд сучасних методів стеганографії, окреслено переваги та недоліки кожного з них. Визначено проблематику, шляхи і методи покращення вже існуючих рішень. Також було поставлено задачу для виконання да досягнення остаточної мети даної магістерської роботи, а саме вдосконалення багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Також було здійснено розробку і проектування ключових алгоритмів роботи вдосконаленого багатоканального стеганографічного методу з динамічним розподілом інформації, для програмної реалізації. Було розроблено наступні алгоритми роботи: загальний алгоритм роботи під час приховування інформації, алгоритм динамічного розподілу інформації між носіями на основі даних і властивостей носіїв та алгоритм шифрування розділених частин повідомлення перед приховуванням для кращого захисту інформації на основі введеного користувачем пароля та згенерованого ключа на основі пароля.

Також було здійснено проектування інтерфейсу для програмної реалізації вдосконаленого багатоканального стеганографічного методу з динамічним розподілом інформації. При розробці і проектуванні інтерфейсу було враховано і пропрацьовано все для максимальної зручності в користуванні майбутнім розробленим застосунком.

Було обґрунтовано вибір мови програмування та середовища розробки, окреслено недоліки та ключові переваги обраних рішень. Також було здійснено саму програмну реалізацію вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації, детально розписано процес розробки.

Також проведено тестування програмно реалізовано вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації. Проведено наступні тестування:

- Тестування на продуктивність та масштабованість;
- Тестування динамічного розподілу;
- Тестування цілісності повідомлення після приховування та отримання у зворотньому порядку;
- Оцінка якості носіїв після вбудовування;
- Тестування криптографічної стійкості;
- Можливість одноканального приховування.

Після чого також було проведено оцінку комерційного потенціалу вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації. Оцінку було проведено за 12-ма пунктами по 5-ти бальні шкалі, в результаті сформовано відповідні висновки. Проведено розрахунок собівартості розробки з урахуванням витрат пов’язаних з проведенням науково-дослідної роботи які групуються за наступними статтями:

- основна заробітна плата виконавців робіт;
- додаткова заробітна плата;
- матеріальні витрати (податки);
- витрати на спеціальне обладнання для наукових (експериментальних) робіт;
- амортизаційні відрахування;
- програмне забезпечення для наукових робіт;
-інші витрати.

В результаті було отримано загальний прогнозований бюджет розробки вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації. Також розраховано

коефіцієнт економічної ефективності та період окупності вкладених інвестицій інвестором.

В результаті роботи над даною дипломною роботою сформовано висновки що вдосконалений багатоканальний стеганографічний метод з використанням динамічного розподілу прихованої інформації, є дійсно вдосконаленим і ефективнішим, кращим в порівнянні з існуючими версіями багатоканальних стеганографічних методів, це підтверджують результати тестувань.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Тези доповіді для Міжнародної науково-практичної Інтернет-конференції студентів, аспірантів та молодих науковців «Молодь у науці: дослідження, проблеми, перспективи (МН-2026)» URL: https://conferences.vntu.edu.ua/index.php/mn/mn2026/author/submission/26683 (Дата звернення: 04.12.2025.).

2. Різниця між криптографією та стеганографією. URL: https://hackyourmom.com/osvita/shho-take-steganografiya-prynczypy-ta-zastosuvannya-v-kiberbezpeczi/ (Дата звернення: 07.09.2025.).

3. Поняття криптографії. URL: https://sites.google.com/view/blog-ua (Дата звернення: 07.09.2025.).

4. Криптографічний алгоритм. URL: https://sites.google.com/view/blog-ua (Дата звернення: 07.09.2025.).

5. Особливості стеганографії. URL: стеганографії. https://www.geeksforgeeks.org/computer-networks/what-is-steganography/ (Дата звернення: 07.09.2025.).

6. Стеганографія. URL:https://www.digitalregenesys.com/blog/what-is-steganography (Дата звернення: 07.09.2025.).

7. Історія появи стеганографії. URL: https://studfile.net/preview/5171438/page:3/ (Дата звернення: 07.09.2025).

8. Acrostic Linguistic Steganography. URL: https://ieeexplore.ieee.org/document/9714779/ (Дата звернення: 07.09.2025).

9. Найпростіші методи застосування стеганографії URL: https://studfile.net/preview/5171438/page:3/ (Дата звернення: 07.09.2025.).

10. Мельник С. МЕТОДИ ЦИФРОВОЇ СТЕГАНОГРАФІЇ: СТАН ТА НАПРЯМИ РОЗВИТКУ // С. Мельник, В. Кащук. // Information Security of the Person, Society and State. – 2013. – №3. – С. 65–70;

11. Класична стеганографія URL: https://er.knutd.edu.ua/bitstream/123456789/19934/1/SEIS_mono_2021_P090-094.pdf (Дата звернення: 07.09.2025.).

12. Принцип дії класичної стеганографії URL: https://er.knutd.edu.ua/bitstream/123456789/19934/1/SEIS_mono_2021_P090-094.pdf (Дата звернення: 07.09.2025.).

13. Computer steganography. URL: https://ijigroup.com/en/stenography-3/ (Дата звернення: 07.09.2025.).

14. Цифрова стеганографія URL: https://er.knutd.edu.ua/bitstream/123456789/19934/1/SEIS_mono_2021_P090-094.pdf (Дата звернення: 07.09.2025.).

15. Принцип методу реалізації комп’ютерної стеганографії. URL: https://repository.hneu.edu.ua/bitstream/ (Дата звернення: 07.09.2025.).

16. Типи файлів у сучасній комп’ютерній стеганографії. URL: https://www.e3s-conferences.org/articles/e3sconf/pdf/2024/68/e3sconf_ipfa2024_02022.pdf (Дата звернення: 07.09.2025.).

17. Навроцький Д.О. ЗАСОБИ ЗАХИСТУ ІНФОРМАЦІЇ - МЕТОДИ КОМП'ЮТЕРНОЇ СТЕГАНОГРАФІЇ - 35ст.

18. Базові положення сучасної комп’ютерної стеганографії. URL: https://www.academia.edu/7069617/Principles_of_Modern_Steganography_and_Steganalysis (Дата звернення: 07.09.2025).

19. Різниця між стеганографічними методами. URL: https://www.sciencedirect.com/topics/computer-science/steganographic-technique (Дата звернення: 07.09.2025).

20. Непомітність як один із найважливіших аспектів оцінки якості методу стеганографічного приховування інформації. URL: https://ceur-ws.org/Vol-4024/paper07.pdf (Дата звернення: 07.09.2025).

21. Ємність в сучасних методах комп’ютерної стеганографії. URL: https://www.sciencedirect.com/science/article/pii/S2666449624000033 (Дата звернення: 07.09.2025).

22. Важливість стійкості в методах комп’ютерної стеганографії. URL: https://arxiv.org/html/2312.01284v2 (Дата звернення: 07.09.2025).

23. Основні принципи методів стеганографії. URL: https://hackyourmom.com/en/osvita/shho-take-steganografiya-prynczypy-ta-zastosuvannya-v-kiberbezpeczi/ (Дата звернення: 07.09.2025).

24. Огляд стеганографічного методу найменш значущого біту (Least Significant Bit). URL: https://www.techscience.com/cmc/v81n1/58294/html (Дата звернення: 07.09.2025.)

25. Переваги та недоліки стеганографічного методу найменш значущого біту (Least Significant Bit). URL: https://www.sciencedirect.com/org/science/article/pii/S1546221824007501 (Дата звернення: 07.09.2025.)

26. Огляд стеганографічного методу дискретного косинусного перетворення Discrete Cosine Transform (DCT). URL: https://www.researchgate.net/publication/330565811_Hiding_data_in_images_using_DCT_steganography_techniques_with_compression_algorithms (Дата звернення: 07.09.2025.)

27. Принцип дії стеганографічного методу дискретного косинусного перетворення Discrete Cosine Transform (DCT). URL: https://link.springer.com/article/10.1007/s11128-023-03914-5 (Дата звернення: 07.09.2025.)

28. Алгоритм роботи стеганографічного методу (Discrete Cosine Transform DCT). URL: https://ijrpr.com/uploads/V4ISSUE5/IJRPR13625.pdf (Дата звернення: 07.09.2025.)

29. Огляд стеганографічної концепції (Multi-image steganography). URL: https://link.springer.com/article/ (Дата звернення: 07.09.2025.)

30. Недоліки стеганографічної концепції (Multi-image steganography). URL: https://arxiv.org/html/2410.10117v1 (Дата звернення: 07.09.2025.)

31. Порівняльна характеристика описаних в роботі методів комп'ютерної стеганографії. URL: https://www.academia.edu/91167346/Comparative_Study_of_Steganography_Tools (Дата звернення: 07.09.2025.)

32. Multichannel Steganography. URL: https://arxiv.org/html/2501.04511v1 (Дата звернення: 07.09.2025.);

33. Пропускна здатність каналу та формула її визначення. URL: https://www.sciencedirect.com/topics/computer-science/channel-capacity (Дата звернення: 03.09.2025.);

34. Методи покращення стійкості алгоритму до стегааналузу. URL: https://www.researchgate.net/publication/333226568_Deep_Learning_Applied_to_Steganalysis_of_Digital_Images_A_Systematic_Review (Дата звернення: 05.09.2025.);

35. Базові засади багатоканальних методів комп'ютерної стеганографі. URL: https://www.techrxiv.org/users/939735/articles/1357418-multi-channel-linguistic-steganography-using-frequency-multiplexing-and-minimum-entropy-coupling (Дата звернення: 09.09.2025.);

36. Порівняння найпопулярніших методів реалізації комп'ютерної стеганографії. URL: https://mej.researchcommons.org/cgi/viewcontent.cgi?article=2617&context=home (Дата звернення: 11.09.2025.);

37. Відео стеганографія. URL: https://link.springer.com/article/10.1007/s11042-023-14844-w (Дата звернення: 11.09.2025.);

38. Текстова комп’ютерна стеганографія. URL: https://ieeexplore.ieee.org/document/8085643 (Дата звернення: 11.09.2025.);

39. Multi-image steganography. URL: https://arxiv.org/abs/2101.00350 (Дата звернення: 07.09.2025.).

40. Онлайн ресурс для побудови і їх схематичного проедставлення: Lucidchart. URL: https://lucid.app/ (Дата звернення: 07.09.2025).

41. Шифрування. Види шифрування даних та алгоритми їх роботи. URL: https://www.geeksforgeeks.org/dsa/dsa-tutorial-learn-data-structures-and-algorithms/ (Дата звернення: 15.09.2025).

42. Офіційний сайт JavaScript. URL: https://www.javascript.com/ (Дата звернення: 17.09.2025).

43. Переваги під час вибору мови програмування JavaScript. URL: https://www.geeksforgeeks.org/javascript/advantages-and-disadvantages-of-javascript/ (Дата звернення: 19.09.2025).

44. Документація JavaScript. URL: https://developer.mozilla.org/ru/docs/Web/JavaScript (Дата звернення: 24.09.2025).

45. Офіційний сайт середовища розробки Visual Studio Code (VS Code). URL: https://code.visualstudio.com/ (Дата звернення: 07.09.2025).

46. Переваги під час вибору середовища розробки Visual Studio Code. URL: https://www.hostinger.com/tutorials/what-is-vs-code (Дата звернення: 28.09.2025).

47. HTML. URL: https://html.com/ (Дата звернення: 29.09.2025).

48. Програмування на JavaScript. URL: https://www.w3schools.com/js/DEFAULT.asp (Дата звернення: 30.09.2025).

49. Базова структура програмно реалізованого методу компютрної стеганографії LSB. URL: https://github.com/metallurgical/LSB-steganography-javascript (Дата звернення: 25.09.2025).

50. Сайт CSS. URL: https://www.w3.org/Style/CSS/articlelist.uk.html (Дата звернення: 11.10.2025).

51. Синтаксис CSS. URL: https://developer.mozilla.org/en-US/docs/Web/CSS (Дата звернення: 08.10.2025).

52. Ресурс для аналізу аудіо файлів URL: https://editor.audio/uk/ (Дата звернення: 07.09.2025).

53. Оцінка комерційного потенціалу розробки. URL: https://www.researchgate.net/publication/26496753_Measuring_Commercial_Potential_of_a_New_Technology_at_the_Early_Stage_of_Development_with_Fuzzy_Logic (Дата звернення: 09.10.2025).

54. Проведення технологічного аудиту продукту. URL: https://dl.acm.org/doi/10.1007/s10664-014-9358-0 (Дата звернення: 17.10.2025).

55. Оцінки критеріїв комерційного потенціалу вдосконаленого стеганографічного методу з використанням динамічного розподілу прихованої інформації. URL: https://parallelstaff.com/how-to-evaluate-software-development-services/ (Дата звернення: 07.09.2025).

56. Критерії які враховуються під час розрахунку собівартості розробки. URL: https://startups.epam.com/blog/guide-on-software-project-cost-estimation (Дата звернення: 23.10.2025).

57. Розрахунок заробітної плати працівникам залученим до розробки. URL: https://biosistemika.com/blog/estimate-costs-of-software-development-projects/ (Дата звернення: 27.10.2025).

58. Розрахунок спеціального обладнання. URL: https://www.techmagic.co/blog/ai-development-cost (Дата звернення: 01.11.2025).

59. Розрахунок амортизації ТМЦ. URL: https://scispace.com/pdf/cost-and-complexity-research-of-software-development-to-1xtz35g5j1.pdf (Дата звернення: 04.11.2025).

60. Розрахунок економічної ефективності. URL: https://scispace.com/pdf/the-scientific-and-technical-evaluation-of-the-effectiveness-1eog1ry3z8.pdf (Дата звернення: 05.11.2025).

61. Розрахунок ефективності вкладених інвестицій та періоду їх окупності. URL: https://www.investopedia.com/terms/p/paybackperiod.asp (Дата звернення: 07.11.2025).

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
"24" вересня 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:

Вдосконалення багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації

08-72.МКР.004.15.97.ТЗ

Керівник магістерської кваліфікаційної роботи
Карпінець В.В., к.т.н., доцент,
завідувач кафедри МБІС

Вінниця – 2025 р.

1. Найменування та область застосування

Програмний засіб вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації.

Область застосування: забезпечення конфіденційності, цілісності та прихованості інформації у системах кібербезпеки, захисту інформаційних ресурсів, спеціалізованих комунікаційних системах, а також у сферах, де вимагається приховане передавання даних.

2.Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ від 24 вересня 2025 року № 313;

3. Мета та призначення розробки

3.1 та відповідно до навчального плану підготовки магістрів за спеціальністю 125 «Кібербезпека»;

3.2 Призначення: Розроблений програмний засіб призначений для приховування інформації одночасно в декілька мультимедійних контейнерів із забезпеченням підвищеної ємності та стійкості до стеганографічного аналізу та можливістю адаптивного (динамічного) розподілу секретних даних між кількома каналами одночасно;

4. Джерела розробки

4.1. Різниця між криптографією та стеганографією. URL: https://hackyourmom.com/osvita/shho-take-steganografiya-prynczypy-ta-zastosuvannya-v-kiberbezpeczi/.

4.2. Особливості стеганографії. URL: https://www.geeksforgeeks.org/computer-networks/what-is-steganography/.

4.3. Мельник С., Кащук В. Методи цифрової стеганографії: стан та напрями розвитку // Information Security of the Person, Society and State. – 2013. – №3. – С. 65–70.

4.4. Multichannel Steganography. URL: https://arxiv.org/html/2501.04511v1.

4.5. Базові засади багатоканальних методів комп'ютерної стеганографії. URL: https://www.techrxiv.org/users/939735/articles/1357418-multi-channel-linguistic-steganography-using-frequency-multiplexing-and-minimum-entropy-coupling.

4.6. Методи покращення стійкості алгоритму до стегоаналізу. URL: https://www.researchgate.net/publication/333226568_Deep_Learning_Applied_to_Steganalysis_of_Digital_Images_A_Systematic_Review.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен забезпечувати можливість приховування інформації у декількох типах контейнерів (зображення та аудіо);

5.1.2 Реалізація методу повинна включати механізм динамічного розподілу даних між каналами на основі властивостей контейнерів;

5.1.3 Інтерфейс користувача має бути інтуїтивно зрозумілим та зручним;

5.1.4 Програмний засіб не повинен вимагати спеціальних ліцензійних компонентів;

5.1.5 В програмному продукті має бути реалізована можливість як вбудовування, так і вилучення прихованої інформаці у зворотньому порядку;

5.2 Вимоги до надійності:

5.2.1 Програмний засіб має працювати без критичних збоїв; у разі помилок повинні виводитися відповідні повідомлення;

5.2.2 Має бути забезпечена і збережена цілісність як контейнера, так і прихованого повідомлення;

5.2.3 Програмний засіб повинен виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів:

– процесор – Pentium 1500 МГц і подібні до них;

– оперативна пам’ять – не менше 512 Mb;

– середовище функціонування – операційна система сімейство Windows;

– вимоги до техніки безпеки: відповідність чинним нормам при роботі з комп’ютерною технікою.

6. Вимоги до програмної документації

6.1 Документація повинна містити опис алгоритмів та структури даних, що використовуються у програмі.

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист розроблюваного програмного засобу від несанкціонованого використання.

7.2 Використання криптографічних механізмів для додаткового захисту секретного повідомлення.

8. Техніко-економічні показники

8.1 Результати використання розробленого методу повинні перевищувати витрати на його реалізацію за рахунок підвищеної стійкості та багатоканального характеру;

8.2. Програмний засіб має бути достатньо універсальним для використання в широкому спектрі застосувань, від наукових досліджень до практичних систем кіберзахисту.

8.3. Впровадження розробки не потребує дороговартісного обладнання чи спеціального ПЗ.

9. Стадії та етапи розробки

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Початок	Закінчення
1	Визначення напрямку магістерської кваліфікаційної роботи, формулювання теми	23.05.2025	04.09.2025

2	Аналіз предметної області обраної теми	04.09.2025	12.09.2025
3	Аналіз існуючих методів стеганографії	13.09.2025	19.09.2025
4	Визначення і обґрунтування вдосконалення багатоканального стеганографічного методу	19.09.2025	26.09.2025
5	Розробка запропонованих алгоритмів роботи	26.09.2025	09.10.2025
6	Вибір мови програмування та середовища розробки	09.10.2025	10.10.2025
7	Програмна реалізація вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації	10.10.2025	05.11.2025
8	Проведення тестування розробленого вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації	05.11.2025	12.11.2025
9	Здійснення оцінки комерційного потенціалу	13.11.2025	15.11.2025
10	Здійснення аналізу прогнозування витрат на виконання науково-дослідної роботи	16.11.2025	16.11.2025
11	Виконання розрахунку економічної ефективності науково-дослідної роботи	17.11.2025	17.11.2025
12	Здійснення розрахунку ефективності вкладених інвестицій та періоду їх окупності.	18.11.2025	19.11.2025
13	Попередній захист магістерської кваліфікаційної роботи	19.11.2025	19.11.2025
14	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	19.11.2025	04.12.2025
15	Захист магістерської кваліфікаційної роботи	08.12.2025	08.12.2025

10. Порядок контролю та прийому

10.1 До приймання магістерської кваліфікаційної роботи надається:

– ПЗ до магістерської кваліфікаційної роботи;

– програмний додаток;

– презентація;

– відгук керівника роботи;

– відгук опонента

Технічне завдання до виконання прийняв _____ Волос В.С.

Додаток Б. Лістинг програми (js)

```
const fileImage = document.getElementById('fileImage');
const fileAudio = document.getElementById('fileAudio');
const canvas = document.getElementById('canvas');
const ctx = canvas.getContext('2d');
const messageEl = document.getElementById('message');
const encodeBtn = document.getElementById('encodeBtn');
const decodeBtn = document.getElementById('decodeBtn');
const downloadImageBtn = document.getElementById('downloadImageBtn');
const downloadAudioBtn = document.getElementById('downloadAudioBtn');
const info = document.getElementById('info');
const audioInfo = document.getElementById('audioInfo');
const passwordEl = document.getElementById('password');

const imageArea = document.getElementById('imageArea');

let originalImage = null;
let originalAudioBuffer = null;
let originalAudioFile = null;
let audioSampleRate = 44100;
let lastImageBitsUsed = 0;
let lastAudioBitsUsed = 0;


function bitsFromNumber(n, bitsCount) {
 const out = [];
 for(let i=bitsCount-1;i>=0;i--) out.push((n >> i) & 1);
 return out;
}
function bitsFromBytes(byteArray) {
 const out = [];
 for (let b of byteArray) for (let i = 7; i >= 0; i--) out.push((b >> i) & 1);
 return out;
}
function bytesFromBits(bits) {
 const bytes = [];
 for (let i = 0; i < bits.length; i += 8) {
  let val = 0;
  for (let j = 0; j < 8 && i+j < bits.length; j++) val = (val << 1) | bits[i+j];
  bytes.push(val);
 }
 return new Uint8Array(bytes);
}


async function deriveKey(password, salt) {
 const enc = new TextEncoder();
 const pwKey = await crypto.subtle.importKey('raw', enc.encode(password), {name: 'PBKDF2'},
false, ['deriveKey']);
  return crypto.subtle.deriveKey(
   { name: 'PBKDF2', salt: salt, iterations: 200000, hash: 'SHA-256' },
   pwKey,
   { name: 'AES-GCM', length: 256 },
   false,
```

```
    ['encrypt','decrypt']
   );
  }
  async function encryptBytes(password, plainBytes) {
   const salt = crypto.getRandomValues(new Uint8Array(16));
   const iv = crypto.getRandomValues(new Uint8Array(12));
   const key = await deriveKey(password, salt);
   const ct = await crypto.subtle.encrypt({ name: 'AES-GCM', iv: iv }, key, plainBytes);
   return { salt, iv, ciphertext: new Uint8Array(ct) };
  }
  async function decryptBytes(password, salt, iv, ciphertext) {
   const key = await deriveKey(password, salt);
   try {
    const pt = await crypto.subtle.decrypt({ name: 'AES-GCM', iv: iv }, key, ciphertext);
    return new Uint8Array(pt);
   } catch (e) {
    return null;
   }
  }

  function interleaveChannels(channels) {
   const length = channels[0].length;
   const numChannels = channels.length;
   const out = new Int16Array(length * numChannels);
   for (let i = 0; i < length; i++) {
    for (let ch = 0; ch < numChannels; ch++) {
     let s = channels[ch][i];
     s = Math.max(-1, Math.min(1, s));
     out[i * numChannels + ch] = Math.round(s * 32767);
    }
   }
   return out;
  }
  function writeWav(int16Samples, sampleRate, numChannels) {
   const bytesPerSample = 2;
   const blockAlign = numChannels * bytesPerSample;
   const byteRate = sampleRate * blockAlign;
   const dataSize = int16Samples.length * bytesPerSample;
   const buffer = new ArrayBuffer(44 + dataSize);
   const dv = new DataView(buffer);
   let p = 0;
   function writeString(s) { for (let i=0;i<s.length;i++) dv.setUint8(p++, s.charCodeAt(i)); }
   writeString('RIFF'); dv.setUint32(p, 36 + dataSize, true); p += 4; writeString('WAVE');
writeString('fmt '); dv.setUint32(p,16,true); p+=4; dv.setUint16(p,1,true); p+=2;
dv.setUint16(p,numChannels,true); p+=2; dv.setUint32(p,sampleRate,true); p+=4;
dv.setUint32(p,byteRate,true); p+=4; dv.setUint16(p,blockAlign,true); p+=2; dv.setUint16(p,16,true); p+=2;
writeString('data'); dv.setUint32(p,dataSize,true); p+=4;
   for (let i=0;i<int16Samples.length;i++,p+=2) dv.setInt16(p, int16Samples[i], true);
   return new Blob([buffer], {type: 'audio/wav'});
  }


  function splitPlaintext(plainBytes, imageBits, audioBits) {
   const totalCarrierBits = imageBits + audioBits;
   if (totalCarrierBits === 0) return {firstChunk: new Uint8Array(0), secondChunk: new
Uint8Array(0)};
   const audioRatio = audioBits / totalCarrierBits;
```

```
    const totalBits = plainBytes.length * 8;
    const audioBitsToUse = Math.round(totalBits * audioRatio);
    const audioBytesCount = Math.floor(audioBitsToUse / 8);
    const imageBytesCount = plainBytes.length - audioBytesCount;
    const firstChunk = plainBytes.slice(0, imageBytesCount);
    const secondChunk = plainBytes.slice(imageBytesCount);
    return {firstChunk, secondChunk};
  }


  function updateInfo(payloadPlainLength = 0) {
    const w = canvas.width, h = canvas.height;
    const imageBitsCapacity = originalImage ? w * h * 3 : 0;
    const audioBitsCapacity = originalAudioBuffer ? originalAudioBuffer.length *
originalAudioBuffer.numberOfChannels : 0;

    let infoText = `Характеристика зображення: ${originalImage ? `${w}×${h}px —
${Math.floor(imageBitsCapacity/8)} байт` : '— не завантажено'}.\n`;
    infoText += `Характеристика аудіо: ${originalAudioBuffer ? `${originalAudioBuffer.length}
семплів × ${originalAudioBuffer.numberOfChannels} каналів — ${Math.floor(audioBitsCapacity/8)}
байт` : '— не завантажено'}.\n`;

    if (payloadPlainLength > 0) {
      const totalBitsUsed = lastImageBitsUsed + lastAudioBitsUsed;
      const imgPercent = totalBitsUsed === 0 ? 0 :
Math.round((lastImageBitsUsed/totalBitsUsed)*100);
      const audPercent = totalBitsUsed === 0 ? 0 :
Math.round((lastAudioBitsUsed/totalBitsUsed)*100);


      const totalMessageBits = payloadPlainLength * 8;

      infoText += `\nПовідомлення: ${payloadPlainLength} байт (${totalMessageBits} біт)\n`;
      infoText += `Приховано бітів: Image = ${lastImageBitsUsed}, Audio =
${lastAudioBitsUsed}\nРозподіл: Image ${imgPercent}%, Audio ${audPercent}%`;
    } else {
      infoText += `\nВведіть повідомлення і натисніть "Закодувати".`;
    }
    info.textContent = infoText;
  }


  fileImage.addEventListener('change', e => {
    const f = e.target.files[0];
    if(!f) return;
    const img = new Image();
    img.onload = () => {
      canvas.width = img.width;
      canvas.height = img.height;
      ctx.drawImage(img,0,0);
      originalImage = img;
      downloadImageBtn.disabled = true;
      updateInfo();

      const imageBitsCapacity = canvas.width * canvas.height * 3;
    console.log("Image capacity (bytes):", Math.floor(imageBitsCapacity/8));
    };
```

```
  img.src = URL.createObjectURL(f);
});

const audioPreview = document.getElementById('audioPreview');

const audioEl = audioPreview.querySelector('audio');

fileAudio.addEventListener('change', async e => {
  const f = e.target.files[0];
  if (!f) return;
  originalAudioFile = f;


  audioEl.src = URL.createObjectURL(f);
  audioEl.load();
  audioEl.play();
  audioPreview.style.display = 'block';



  const ab = await f.arrayBuffer();
  try {
    const buf = await audioCtx.decodeAudioData(ab.slice(0));
    originalAudioBuffer = buf;
    audioSampleRate = buf.sampleRate;
    audioInfo.textContent = `Audio: ${buf.length} samples, ${buf.numberOfChannels} channels,
${buf.sampleRate} Hz.`;
    downloadAudioBtn.disabled = true;
    updateInfo();
  } catch (err) {
    audioInfo.textContent = 'Не вдалося декодувати аудіо.';
    originalAudioBuffer = null;
    updateInfo();
  }
});


function embedToImage(imageData, bytes) {
  const data = imageData.data;
  const w = imageData.width, h = imageData.height;
  const totalBits = w*h*3;
  const neededBits = (4 + bytes.length) * 8;
  if (neededBits > totalBits) throw new Error('Image capacity too small for given chunk');
  const lenBits = bitsFromNumber(bytes.length >>> 0, 32);
  const msgBits = bitsFromBytes(bytes);
  const bits = lenBits.concat(msgBits);
  let bitIdx = 0;
  for (let px = 0; px < w*h && bitIdx < bits.length; px++) {
    const base = px*4;
    for (let c = 0; c < 3 && bitIdx < bits.length; c++) {
      data[base+c] = (data[base+c] & 0xFE) | bits[bitIdx++];
    }
  }
  lastImageBitsUsed = bitIdx;
}
```

```
        const audioCtx = new (window.AudioContext || window.webkitAudioContext)();
        fileAudio.addEventListener('change', async e => {
         const f = e.target.files[0];
         if(!f) return;
         originalAudioFile = f;
         const ab = await f.arrayBuffer();
         try {
          const buf = await audioCtx.decodeAudioData(ab.slice(0));
          originalAudioBuffer = buf;
          audioSampleRate = buf.sampleRate;
          audioInfo.textContent = `Audio: ${buf.length} samples, ${buf.numberOfChannels} channels,
${buf.sampleRate} Hz.`;
          downloadAudioBtn.disabled = true;
          updateInfo();
         } catch (err) {
          audioInfo.textContent = 'Не вдалося декодувати аудіо.';
          originalAudioBuffer = null;
          updateInfo();
         }
        });


        function embedToAudio(audioBuffer, bytes) {
         const numChannels = audioBuffer.numberOfChannels;
         const length = audioBuffer.length;
         const totalBits = length * numChannels;
         const neededBits = (4 + bytes.length) * 8;
         if (neededBits > totalBits) throw new Error('Audio capacity too small for given chunk');

         const channels = [];
         for (let ch = 0; ch < numChannels; ch++) {
          const floatArr = audioBuffer.getChannelData(ch);
          const intArr = new Int16Array(length);
          for (let i=0;i<length;i++) intArr[i] = Math.round(Math.max(-1, Math.min(1, floatArr[i])) * 32767);
          channels.push(intArr);
         }

         const interleaved = new Int16Array(length * numChannels);
         for (let i = 0; i < length; i++) for (let ch=0; ch<numChannels; ch++)
interleaved[i*numChannels+ch] = channels[ch][i];

         const lenBits = bitsFromNumber(bytes.length >>> 0, 32);
         const msgBits = bitsFromBytes(bytes);
         const bits = lenBits.concat(msgBits);

         for (let i = 0; i < bits.length; i++) {
          interleaved[i] = (interleaved[i] & 0xFFFE) | bits[i];
         }
         lastAudioBitsUsed = bits.length;


         const outChannels = [];
         const samplesPerChannel = interleaved.length / numChannels;
         for (let ch=0; ch<numChannels; ch++) {
          const floatArr = new Float32Array(samplesPerChannel);
          for (let i=0;i<samplesPerChannel;i++) {
           floatArr[i] = interleaved[i*numChannels + ch] / 32767;
```

```
    }
    outChannels.push(floatArr);
   }

   return {int16Interleaved: interleaved, channelsFloat: outChannels};
  }


  function extractFromImage(imageData) {
   const data = imageData.data;
   const w = imageData.width, h = imageData.height;
   const bits = [];
   for (let px = 0; px < w*h; px++) {
    const base = px*4;
    for (let c = 0; c < 3; c++) bits.push(data[base+c] & 1);
   }


   let len = 0;
   if (bits.length < 32) return new Uint8Array(0);
   for (let i = 0; i < 32; i++) len = (len << 1) | bits[i];
   const totalNeeded = 32 + len*8;
   if (bits.length < totalNeeded) {

    const availableBits = Math.max(0, bits.length - 32);
    const partialBytes = Math.floor(availableBits / 8);
    const msgBits = bits.slice(32, 32 + partialBytes*8);
    return bytesFromBits(msgBits);
   }
   const msgBits = bits.slice(32, 32 + len*8);
   return bytesFromBits(msgBits);
  }

  function extractFromAudioBuffer(audioBuffer) {
   const numChannels = audioBuffer.numberOfChannels;
   const length = audioBuffer.length;
   const interleaved = new Int16Array(length * numChannels);
   for (let ch=0; ch<numChannels; ch++) {
    const floatArr = audioBuffer.getChannelData(ch);
    for (let i=0;i<length;i++) interleaved[i*numChannels+ch] = Math.round(Math.max(-1,
Math.min(1, floatArr[i])) * 32767);
   }
   const bits = [];
   for (let i=0;i<interleaved.length;i++) bits.push(interleaved[i] & 1);
   if (bits.length < 32) return new Uint8Array(0);
   let len = 0;
   for (let i = 0; i < 32; i++) len = (len << 1) | bits[i];
   const totalNeeded = 32 + len*8;
   if (bits.length < totalNeeded) {
    const availableBits = Math.max(0, bits.length - 32);
    const partialBytes = Math.floor(availableBits / 8);
    const msgBits = bits.slice(32, 32 + partialBytes*8);
    return bytesFromBits(msgBits);
   }
   const msgBits = bits.slice(32, 32 + len*8);
   return bytesFromBits(msgBits);
  }
```

```
const w = canvas.width, h = canvas.height;
const imageData = ctx.getImageData(0,0,w,h);
const bits = [];
for(let px=0; px<w*h; px++){
 const base = px*4;
 for(let c=0; c<3; c++) bits.push(imageData.data[base+c]&1);
}
console.log(bits.slice(0, 100));


encodeBtn.addEventListener('click', async () => {
 try {
  const text = messageEl.value;
  if (!text) return alert('Введіть повідомлення.');
  const password = passwordEl.value;
  if (!password) return alert('Введіть пароль.');

  const plainBytes = new TextEncoder().encode(text);

  const w = canvas.width, h = canvas.height;
  const imageBits = originalImage ? w*h*3 : 0;
  const audioBits = originalAudioBuffer ? originalAudioBuffer.length *
originalAudioBuffer.numberOfChannels : 0;


  const {firstChunk: firstPlain, secondChunk: secondPlain} = splitPlaintext(plainBytes, imageBits,
audioBits);

  const totalCarrierBits = imageBits + audioBits;
const totalMessageBits = plainBytes.length * 8;

if (totalMessageBits > totalCarrierBits) {
 return alert(` Повідомлення занадто велике для наявних носіїв!\nМаксимум:
${Math.floor(totalCarrierBits/8)} байт, Ваше повідомлення: ${plainBytes.length} байт.`);
}

  let imgEmbedBytes = new Uint8Array(0);
  let audioEmbedBytes = new Uint8Array(0);

  if (firstPlain.length > 0 && originalImage) {
   const {salt, iv, ciphertext} = await encryptBytes(password, firstPlain);
   imgEmbedBytes = new Uint8Array(salt.length + iv.length + ciphertext.length);
   imgEmbedBytes.set(salt, 0);
   imgEmbedBytes.set(iv, salt.length);
   imgEmbedBytes.set(ciphertext, salt.length + iv.length);
  }


  if (secondPlain.length > 0 && originalAudioBuffer) {
   const {salt, iv, ciphertext} = await encryptBytes(password, secondPlain);
   audioEmbedBytes = new Uint8Array(salt.length + iv.length + ciphertext.length);
   audioEmbedBytes.set(salt, 0);
   audioEmbedBytes.set(iv, salt.length);
   audioEmbedBytes.set(ciphertext, salt.length + iv.length);
  }
```

```
            if (originalImage && imgEmbedBytes.length > 0) {
              const imageData = ctx.getImageData(0,0,w,h);
              embedToImage(imageData, imgEmbedBytes);
              ctx.putImageData(imageData,0,0);
              downloadImageBtn.disabled = false;
            }

            if (originalAudioBuffer && audioEmbedBytes.length > 0) {
              const {int16Interleaved} = embedToAudio(originalAudioBuffer, audioEmbedBytes);
              const wavBlob = writeWav(int16Interleaved, audioSampleRate,
originalAudioBuffer.numberOfChannels);
              window._lastStegoAudioBlob = wavBlob;
              downloadAudioBtn.disabled = false;
            }

            if ((!originalImage || imgEmbedBytes.length === 0) && (!originalAudioBuffer ||
audioEmbedBytes.length === 0)) {
              return alert('Немає доступних носіїв або повідомлення розподілено у нуль байт для
доступних носіїв.');
            }

            alert('Приховування завершено. Якщо носій доступний, збережіть стего-файли.');
            lastImageBitsUsed = lastImageBitsUsed || 0;
            lastAudioBitsUsed = lastAudioBitsUsed || 0;
            updateInfo(plainBytes.length);
          } catch (err) {
            console.error(err);
            alert('Помилка: ' + (err && err.message ? err.message : err));
          }
        });

        downloadImageBtn.addEventListener('click', () => {
          canvas.toBlob(blob => {
            const a = document.createElement('a');
            const url = URL.createObjectURL(blob);
            a.href = url; a.download = 'stego_image.png'; a.click(); URL.revokeObjectURL(url);
          }, 'image/png');
        });

        downloadAudioBtn.addEventListener('click', () => {
          const blob = window._lastStegoAudioBlob;
          if (!blob) return alert('Немає стего-аудіо.');
          const a = document.createElement('a');
          const url = URL.createObjectURL(blob);
          a.href = url; a.download = 'stego_audio.wav'; a.click(); URL.revokeObjectURL(url);
        });


        decodeBtn.addEventListener('click', async () => {
          try {
            const password = passwordEl.value;
            if (!password) return alert('Введіть пароль для розкодування.');

            let imagePartBytes = new Uint8Array(0);
            let audioPartBytes = new Uint8Array(0);
```

```
            if (originalImage) {
              const w = canvas.width, h = canvas.height;
              const imageData = ctx.getImageData(0,0,w,h);
              imagePartBytes = extractFromImage(imageData);
            }

            if (originalAudioBuffer) {
              audioPartBytes = extractFromAudioBuffer(originalAudioBuffer);
            }

            let partsText = [];

            async function tryDecryptPart(bytes) {
              if (!bytes || bytes.length < 28) return null;
              const salt = bytes.slice(0,16);
              const iv = bytes.slice(16,28);
              const ciphertext = bytes.slice(28);
              const dec = await decryptBytes(password, salt, iv, ciphertext);
              if (dec === null) return null;
              return new TextDecoder().decode(dec);
            }

            const imgText = await tryDecryptPart(imagePartBytes);
            const audText = await tryDecryptPart(audioPartBytes);

            if (imgText !== null) partsText.push(imgText);
            if (audText !== null) partsText.push(audText);

            if (partsText.length === 0) {
              const hadAny = (imagePartBytes && imagePartBytes.length > 0) || (audioPartBytes &&
audioPartBytes.length > 0);
              if (hadAny) return alert('Наявні дані знайдено, але розшифрування не вдалося
(неправильний пароль або урізані дані).');
              return alert('Немає даних для декодування у завантажених файлах.');
            }

            const finalText = partsText.join('');
            messageEl.value = finalText;
            alert('Видобуток інформації завершено. Показано доступні частини повідомлення.');
          } catch (err) {
            console.error(err);
            alert('Помилка декодування: ' + (err && err.message ? err.message : err));
          }
        });
```

Додаток В. Клієнтський код (html)

```
<!doctype html>
<html lang="uk">
<head>
 <meta charset="utf-8" />
 <meta name="viewport" content="width=device-width,initial-scale=1" />
 <title>Вдосконалений багатоканальний стеганографічний метод з динамічним розподілом інформації</title>
 <link rel="stylesheet" href="style.css">
</head>
<body>
 <h1>Вдосконалений багатоканальний стеганографічний метод з динамічним розподілом інформації</h1>
 <div class="card upload" id="imageUpload">
  <label>1) Завантажити зображення (PNG або JPG)</label>
  <div class="image-row">
   <div class="upload-area" id="imageArea">
    <input id="fileImage" type="file" accept="image/*">
    <span>Натисніть, щоб вибрати зображення</span>
   </div>
   <canvas id="canvas" width="400" height="300"></canvas>
  </div>
 </div>
 <div class="card upload" id="audioUpload">
  <label>2) Завантажити аудіофайл</label>
  <div class="audio-row">
   <div class="upload-area" id="audioArea">
    <input id="fileAudio" type="file" accept="audio/*">
    <span>Натисніть, щоб вибрати аудіо</span>
   </div>
   <div id="audioPreview">
    <audio controls></audio>
   </div>
  </div>
  <div id="audioInfo" class="muted"></div>
 </div>
 <div class="card">
  <label>3) Пароль (ключ для шифрування)</label>
  <input id="password" type="password" placeholder="Введіть пароль...">
  <label>4) Повідомлення для приховування</label>
  <textarea id="message" placeholder="Введіть текст..."></textarea>
  <div class="row">
   <button id="encodeBtn">Приховати</button>
   <button id="decodeBtn">Видобути</button>
   <button id="downloadImageBtn" disabled>Завантажити стего-зображення</button>
   <button id="downloadAudioBtn" disabled>Завантажити стего-аудіо</button>
  </div>
  <div id="info" class="muted" style="margin-top:10px"></div>
 </div>
 <script src="script.js"></script>
</body>
</html>
```

Додаток Г. Ілюстративний матеріал

Вдосконалення багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації

Виконав студент групи 2КІТС-24м Волос Віталій Сергійович
Науковий керівник: к.т.н., доцент каф. МБІС Карпінець Василь Васильович

Актуальність обраної теми

У сучасних умовах стрімкого зростання кіберзагроз та активного перехоплення інформаційних потоків особливої важливості набувають методи прихованої передачі даних. Традиційні стеганографічні підходи часто демонструють обмеження щодо стійкості, пропускної здатності та здатності протистояти методам виявлення. Тому тема магістерської роботи, присвячена вдосконаленню багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації, тема є вкрай актуальною. Запропоноване вдосконалення спрямоване на підвищення надійності, гнучкості та ефективності прихованих каналів у сучасних інформаційних системах.

Мета

Вдосконалити та програмно реалізувати багатоканальний стеганографічний метод із використанням динамічного розподілу прихованої інформації для підвищення стійкості, ефективності та надійності передачі даних у сучасних інформаційних системах.

Завдання

- проаналізувати існуючі стеганографічні методи з визначенням їх переваг та недоліків;

- обґрунтування та аналіз вдосконалення багатоканального стеганографічного методу;

- вибір та обґрунтування стеганографічних каналів у вдосконаленому багатоканальному стеганографічному методі з використанням динамічного розподілу прихованої інформації;

- проектування алгоритму основних алгоритмів роботи вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації;

- програмно реалізувати вдосконалений багатоканальний стеганографічний метод з використанням динамічного розподілу прихованої інформації;

- провести експериментальні дослідження ефективності запропонованих рішень;

- Виконати розрахунки економічної частини розробки.

Аналіз існуючих методів комп’ютерної стеганографії

Критерій оцінки	Least Significant Bit (LSB)	Discrete Cosine Transform (DCT)	Multi-image steganography
Принцип роботи	Приховування інформації в найменш значущих бітах пікселів файлів.	Дані вставляються у коефіцієнти DCT (частотна область)	Повідомлення ділиться на рівні частини, кожна з яких ховається у кожному із зображення.
Тип середовища	Універсальний (BMP, PNG, TIFF, RAW, WAV, AIFF, AVI, MP4)	JPEG	Лише зображення
Видимість змін	Дуже низька (зміни невидимі для людського ока)	Майже непомітні, але при високій інтенсивності зміни можна помітити	Залежить від кількості носіїв, їх розмірів і розмірі приховуваної інформації
Ємність (обсяг даних, які можна сховати)	Висока, дозволяє зберігати великі обсяги даних	Середня, обмежена кількістю коефіцієнтів, придатних до зміни.	Залежить від кількості і розміру зображень-носіїв які використовуються

Обґрунтування та аналіз вдосконалення багатоканального стеганографічного методу.

Класичний метод Multi-image Steganography передбачає рівномірний розподіл повідомлення між кількома носіями, незалежно від їхньої структури, типу або якості. Такий підхід має низку переваг — простоту реалізації, однакове навантаження між каналами та відносно швидку обробку. Проте він не враховує неоднорідність каналів і не забезпечує адаптації системи до зміни їхніх властивостей. У результаті спостерігається надлишкове навантаження на окремі носії, що підвищує ймовірність виявлення прихованої інформації та зменшує загальну стійкість системи.

У вдосконаленому методі запропоновано не лише різнотипні канали приховування (не лише зображення), а принципово інший підхід — динамічний розподіл даних, який здійснюється відповідно до характеристик обраних носіїв (каналів). Завдяки адаптивному підходу метод автоматично підлаштовується під особливості різнотипних носіїв (зображень, аудіо), що суттєво підвищує показники непомітності та стійкості системи до стеганоаналізу. Таким чином, динамічний розподіл забезпечує оптимальний баланс між ємністю, непомітністю та стійкістю, роблячи метод більш універсальним і безпечним у порівнянні з аналогом.

Приклад динамічного розподілу

Співвідношення розподілу		**Початковий розмір повідомлення (біти)**	**Ємність каналів (біти)**		**Розподіл (біти)**	
Зображення (канал 1)	**Аудіо (канал 2)**		**Зображення (канал 1)**	**Аудіо (канал 2)**	**Зображення (канал 1)**	**Аудіо (канал 2)**
100%	**0%**	1000	2000	Канал не обрано	1000	0
75%	**25%**	1000	1500	500	750	250
50%	**50%**	1000	2000	2000	500	500
25%	**75%**	1000	500	1500	250	750
0%	**100%**	1000	Канал не обрано	2000	0	1000

Порівняння методів (існуючого і вдосконаленого)

Критерій оцінки	Існуючий метод	Вдосконалений метод
Тип розподілу	Рівномірний, без урахування параметрів обраних носіїв	Динамічний, пропорційний до ємності обраних носіїв (каналів)
Типи каналів	Лише однотипні (зображення)	Різнотипні (зображення, аудіо, відео)
Ємність системи	Обмежена, часто не використана, або переповнена ємність одного із каналів	Збільшена та максимально ефективна, пропорційно використовується кожен канал
Непомітність	Середня, з ризиком перевищення допустимого спотворення	Вища завдяки адаптивному навантаженню на кожен із каналів
Стійкість до атак	Середня, залежить від кількості каналів	Підвищена за рахунок розподілу за типом і шифрування

З наведеного порівняння можна зробити висновок, що вдосконалений багатоканальний стеганографічний метод з використанням динамічного розподілу інформації має перевагу за всіма ключовими параметрами, хоча є дещо складніший в реалізації.

Вибір та обґрунтування стеганографічних каналів у вдосконаленому багатоканальному стеганографічному методі

Критерії	Зображення	Аудіо	Відео	Текст
Прихована ємність	Висока, великі обсяги даних	Середня, залежить від частоти дискретизації	Дуже висока, багато кадрів + звук	Низька, дуже мало місця
Рівень непомітності	Дуже високий	Високий, слух не помічає низькорівневих змін	Дуже високий	Низький, оскільки зміни можуть бути помітні
Стійкість до атак	Середня, вразливість до статистичних атак	Висока, важче аналізувати акустичні сигнали	Висока, важко аналізувати весь потік	Середня
Вразливість до перетворень	Висока, змінюється при JPEG-компресії	Середня, вразливе до перекодування MP3	Низька, стиснення відео сильно руйнує дані	Дуже висока, будь-яке форматування руйнує дані
Переваги	Простота реалізації, велика ємність	Важко виявити, хороша стійкість	Середня	Легкість передачі, невеликі розміри

З огляду на це було обрано два основних канали такі як цифрове зображення та аудіофайл, які найбільше підходять під потреби завдання.

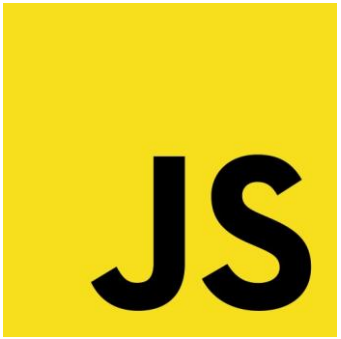
Проектування алгоритмів роботи системи

Загальний алгоритм роботи системи

Алгоритм динамічного розподілу

Обґрунтування вибору мови програмування та середовища розробки.

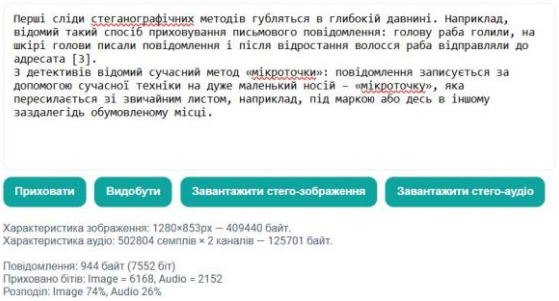
Visual Studio Code



JavaScript

Тестування програмно реалізованого вдосконаленого стеганографічного методу

- Тестування на продуктивність та маштабованість;
- Тестування динамічного розподілу;
- Тестування цілісності повідомлення після приховування та отримання повідомлення у зворотньому порядку;
- Оцінка якості носіїв після приховування;
- Тестування криптографічної стійкості;
- Можливість одноканального приховування.



Тестування і оцінка якості носіїв після приховування

Розташування: C:\Users\vital\Downloads
Розмір: 472 КБ (484 059 байтів)
На диску: 480 КБ (491 520 байтів)

Розташування: C:\Users\vital\Downloads
Розмір: 327 КБ (335 203 байтів)
На диску: 328 КБ (335 872 байтів)

Файли до приховування

Файли після приховування

Розташування: C:\Users\vital\Downloads
Розмір: 1,91 МБ (2 011 260 байтів)
На диску: 1,92 МБ (2 015 232 байтів)

Розташування: C:\Users\vital\Downloads
Розмір: 2,59 МБ (2 725 532 байтів)
На диску: 2,60 МБ (2 727 936 байтів)

Висновки

В результаті виконання магістерської кваліфікаційної роботи на тему «Вдосконалення багатоканального стеганографічного методу з використанням динамічного рзподілу прихованої інформації» було здійснено:

- алізу існуючих стеганографічних методів з визначенням їх переваг та недоліків;
- обґрунтування та аналіз вдосконалення багатоканального стеганографічного методу;
- вибір та обґрунтування стеганографічних каналів у вдосконаленому багатоканальному стеганографічному методі з використанням динамічного розподілу прихованої інформації;
- проектування алгоритму основних алгоритмів роботи вдосконаленого багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації;
- програмно реалізувати вдосконалений багатоканальний стеганографічний метод з використанням динамічного розподілу прихованої інформації;
- проведено експериментальні дослідження ефективності запропонованих рішень;
- виконати економічну частину розробки.

Отже в результаті виконання МКР досягнуто запланованого результату та поставлених цілей, а саме отримано повноцінний, функціональний та програмно реалізований вдосноналений стеганогорафічний метод з використанням динамічного розподілу прихованої інформації, який забезпечує оптимальний баланс між ємністю, непомітністю та захищеністю прихованих даних. Отримані результати мають практичну цінність і можуть бути використані для подальших наукових досліджень або впроваджені у системи прихованого передавання інформації.

Дякую за увагу!

Додаток Д. Протокол перевірки на антиплагіат

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Вдосконалення багатоканального стеганографічного методу з використанням динамічного розподілу прихованої інформації

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра менеджменту та безпеки інформаційних систем
факультет менеджменту та інформаційної безпеки
гр.1КІТС-24м

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism (КП1) **8,78%**

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 _____ доц. Карпінець В.В.

Здобувач _____ Волос В.С.