МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-05

Дипломний проект

здобувача освіти денної форми навчання
РП.05.05.000.ДП

ГОДУЙКА
ДАНІЛА В'ЯЧЕСЛАВОВИЧА

м. Одеса
2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: **121 «Інженерія програмного забезпечення»**
Освітня програма: «**Розробка програмного забезпечення»**
Група: **4РП-05**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка алгоритмічного та програмного забезпечення для стеганофонії

Проектний матеріал складається з пояснювальної записки на 67 сторінках та графічного (презентаційного) матеріалу на 25 аркушах (слайдах).

Дипломник _____ (Годуйко Д.В.)
Керівник _____ (Кривченко А.А.)

Консультанти:
з економічної частини _____ (Копайгородська Т.Г.)
з охорони праці_____ (Чорновол Н.І.)
з дотримання вимог ЄСКД _____ (Петрашова В.І.)
старший консультант _____ (Скорнякова О.В.)

До захисту допущений
Голова циклової комісії _____ (Скорнякова О.В.)
Завідувач відділення _____ (Суліма Ю.Ю.)

Захист «____»_____ 2022 р. Протокол ДКК №_____
Оцінка ДКК _____

Секретар ДКК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення **комп'ютерних систем** Комісія **КТ та ПІ**

Спеціальність **121 «Інженерія програмного забезпечення»**

Освітня програма **«Розробка програмного забезпечення»**

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Беркань І.В.

"_____" _____ 2022 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Годуйку Даніілу В'ячеславовичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка алгоритмічного та програмного забезпечення для стеганофонії

затверджена наказом по коледжу від "____" _____ 202___ р. № _____

2. Термін здачі закінченого проекту (роботи) _____

3. Вихідні данні до проекту (роботи) _____

1. Характеристики аудіо-даних різних типів (мови, музики, складного сигналу)
2. Специфікації методу стеганофонії LSB
3. Специфікації методів стиснення аудіо-даних (OGG, AC3, AAC, WMA, FLAC, APE, ALAC)
4. Забезпечити у алгоритмі стійкість до атак та зменшення спотворення вхідного контейнеру
5. Провести дослідження ефективності модифікованого алгоритму на різних аудіо-файлах

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Модель, структура та характеристики стеганографічної системи;
Огляд методів стеганофонії та вибір оптимального;
Модифікація алгоритму LSB та побудова алгоритмів для ПЗ стеганофонії;
Реалізація програмного продукту на аналіз результатів роботи програми;
Економічна частина; Охорона праці

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Стенографічний алгоритм приховування повідомлення; Структурна схема стегосистеми; Метод LSB для стеганофонії; Алгоритм стиснення інформації; Блок-схема алгоритму приховування даних у аудіо-файлі; Набір вхідних даних; Оцінка спотворення вхідного файлу; Інтерфейс програмного продукту для стеганофонії; Результати процесу приховування даних; Результати додавання АБГШ-атаки; Результати розрахунку показників цілісності; Результати розрахунку показників цілісності для 1-LSB методу

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Технологічний	Кривченко А.А.		
Економічна частина	Копайгородська Т.Г.		
Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Скорнякова О.В.		

7. Дата видачі завдання _____

Керівник Кривченко А.А. _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Вступ. Постановка мети та задач проектування		
2	Аналіз характеристик аудіо-сигналу		
3	Огляд стеганографії, актуальність тематики		
4	Аналітичний огляд та класифікація існуючих		
	методів стеганографії та стеганофонії		
5	Вивчення методів стеганографії аудіо-файлів		
6	Вивчення особливостей алгоритму LSB		
7	Порівняння методів стеганофонії для аудіо-файлів		
8	Опис математичної моделі алгоритму LSB		
9	Опис модифікації методу LSB, розробка алгоритму		
10	Опис засобів розробки ПП, створення ПП		
11	Опис та тестування програмного продукту для модифікації алгоритму LSB		
12	Аналіз результатів, підготовка слайдів презентації		
13	Економічні розрахунки та питання з охорони праці		
14	Підготовка графічної частини проекту		
15	Підготовка проекту до захисту та тестування ПП		

Дипломник _____
(підпис)

Керівник _____
(підпис)

Формат	Зона	Поз.	Позначення	Назва	Кіл.	Примітка
				Документація		
			РП 05. 05 000. 00 ДП	Дипломний проект		
А4			РП 05. 05 000. 00 ДП ПЗ	Пояснювальна записка	1	

Зм.	Арк.	№ докум.	Підпис	Дата	РП 05. 05 000. 00 ДП		
Розробив		Годуйко Д.В.			Розробка алгоритмічного та програмного забезпечення для стеганофонії	Літ.	Аркуш 4 / 67
Перевірив		Кривченко А.А.				Н Д П	
Н. Контр.		Петрашова В.І.				ВСП "ОТФК ОНТУ" гр.4РП-05	
Затверд.		Скорнякова О.В.					

ЗМІСТ

ВСТУП

В Україні та світі найбільшого розвитку здобула така наука про методи забезпечення конфіденційності та автентичності інформації, як криптографія. Разом з тим альтернативний захист може бути створений на базі стеганографії, а в деяких застосуваннях і шляхом використання криптостеганографічних модулів. Стеганографічні методи за своєю природою забезпечують більш високий рівень захисту, оскільки дані, що захищаються, та відповідно, факт їх передачі залишаються поза зоною уваги неуповноважених осіб. Сучасні комп'ютерні технології обробки даних істотно підвищили рівень інформаційної безпеки завдяки глибокій інтеграції криптографічних засобів в інформаційні системи.

На відміну від криптографічного захисту інформації стеганографічні програмні засоби намагаються насамперед приховати сам факт існування конфіденційної інформації. Стеганографічні методи, які приховують інформацію у потоках оцифрованих сигналів та реалізуються на основі комп'ютерної техніки і програмного забезпечення, становлять предмет вивчення цифрової стеганографії.

Цифрова стеганографія приховує сам факт передачі або зберігання інформації, що досягається шляхом впровадження інформації, що захищається, в різні мультимедійні об'єкти (контейнери), які не втрачають від цього своїх споживчих властивостей. У комп'ютерній стеганографії для цього використовуються файли різних форматів, мережеві пакети і т.д. З іншого боку, приховування даних можна використовувати в не комерційному секторі, щоб приховати інформацію, яку хтось хоче зберігати в секреті.

Стеганографія стала доступна для більшості користувачів і може застосовуватися в протизаконних цілях, наприклад, для несанкціонованої передачі комерційних або державних секретів; переписки терористичних угруповань. Тому з'являється необхідність у розробці ефективних методів виявлення прихованих вкладень, в мультимедійних об'єктах, переданих в комп'ютерних мережах. Комп'ютерні технології надали нового імпульсу розвитку

стеганографії, з'явилася комп'ютерна стеганографія, яка забезпечила непомітне, з позицій споживчих якостей, вбудовування даних в файли-контейнери, що містять в цифровому вигляді звуку або зображення. Інтерес до цієї області залишається на високому рівні, хоча вже існує багато застосувань стеганографії на практиці. Прикладами таких застосувань є:

– захист інформації від несанкціонованого доступу;

– протидія системам моніторингу та керуванням ресурсами мереж;

– маскування програмного забезпечення від незареєстрованих користувачів;

– захист авторського права на деякі види інтелектуальної власності [3].

Приховану інформацію можна впровадити в звуковий сигнал, який згодом відтворюється практично точно так (з тією ж якістю), як вхідний сигнал без впровадження. Стеганофонічні системи – це системи передачі звукових повідомлень, у яких приховується факт передачі таємного повідомлення, а саме повідомлення інкапсулюється у стек мережевих протоколів та передається у реальному масштабі часу [2].

Комп'ютерна стеганофонія є достатньо молодою галуззю, яка дозволяє, зокрема, вирішити проблеми пов'язані із захистом авторського права, ідентифікацією та аутентифікацією користувачів.

Для користувачів стеганофонічних систем важливо вибрати оптимальний алгоритм стиснення мовних сигналів та час розмови, за який відбудеться передача прихованого повідомлення. Поточне покоління стеганофонічних аудіо-контейнерів вимагає подальшого удосконалення. Ці поліпшення включають в себе ефективні методи для поліпшення стеганографічної стійкості стеганоконтейнерів. Саме тому, даний дипломний проект присвячено аналізу методів стеганофонії та розробці алгоритмічного і програмного забезпечення для стеганофонії.

1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1.1 Модель стеганографічної системи

Задачею стеганографічної системи є розмістити вхідне повідомлення в контейнері таким чином, щоб будь-яка сторонняя людина не змогла помітити нічого, крім його основного вмісту. Основний вміст контейнера не відіграє ніякої ролі ні для відправника, ні для одержувача, яких цікавить лише успішна передача повідомлення, вміщеного в ньому (стеганограми). Потрібно обов'язково враховувати те, що сам факт відправлення контейнера від автора до одержувача не повинен виглядати дивним, а також не повинно спостерігатись помітних відхилень контейнера від норми.

Узагальнена модель стеганографічної системи схематично представлена на рис. 1.1.

Рисунок 1.1. Узагальнена модель стеганографічної системи

1.2 Основні характеристики стеганографічної системи

Вкраплення повідомлення в контейнер таким чином, щоб будь-який сторонній спостерігач не зміг помітити різниці між оригінальним контейнером та модифікованим, є задачею будь-якої стеганографічної системи. Зазвичай система

будується так щоб забезпечити певний компроміс її базових характеристик, до яких відносяться невідчутність, стійкість, безпека, пропускна здатність створюваного стеганоканалу та обчислювальна складність реалізації.

Невідчутність. Вкраплення повідомлення повинне зберігати перцепційну якість оригінального контейнера. Для аудіосигналів повідомлення повинне бути невідчутним, для зображень – візуально непомітним. Невідчутності повідомлення можна досягнути внесенням мінімальних модифікацій при стеганоперетворенні контейнера, наприклад, на рівні похибки квантування при оцифровці. Крім того, досягти невідчутності допомагає врахування властивостей систем людського слуху та зору. Так, людське вухо працює в режимі частотного аналізатору, що має інтегруючі властивості у межах критичних смуг слуху [6]. Воно здатне сприймати коливання від 20 до 20000 Гц, при цьому найбільш чутливе до звукових компонент з частотами від 500 до 6000 Гц. При розробленні аудіостеганометодів можуть бути використані такі особливості системи людського слуху [5]:

– модифікації, що вносяться в компоненти аудіосигналу, які лежать нижче абсолютного порогу чутності, невідчутні людині;

– поріг чутності одних звукових компонент змінюється в присутності інших: слабке, але чутне звукове коливання стає невідчутним при наявності більш гучного, тобто маскуючим;

– при сприйнятті аудіосигналів людиною крім частотного маскування відбувається також часове, яке ділять на післямаскування та передмаскування.

Числовими показниками невідчутності на практиці часто стають співвідношення сигнал/шум SNR, максимальна різниця MD, середньоквадратична похибка MSE та інші [6].

Стійкість. Суть поняття стійкості залежить від типу атак, які характерні для тієї чи іншої стеганографічної системи. Так, для систем прихованої передачі даних найбільш характерними є пасивні атаки, тому у цьому випадку під стійкою насамперед розуміють систему, яка здатна ефективно їм протидіяти [3].

Стійкість для інших видів стеганосистем, як правило, оцінюють через

кількість помилок, що виникли при вилученні повідомлення з контейнера легальним користувачем після можливих спотворень цього контейнера ненавмисними чи активними атаками. Наприклад, дослідження стійкості до процесів друку та сканування, що обов'язково супроводжують стеганоконтейнер у задачах захисту інформації на паперових носіях.

Застосуванням системи визначається необхідний рівень стійкості. Так, говорячи про стійкість до активних атак, виділяють системи ЦВЗ зі стійкими, крихкими та напівкрихкими водяними знаками [5].

Розглянемо детальніше стійкість у моделях пасивного та активного противників. Стеганосистема та відповідно стеганоконтейнери, які вона продукує, вважаються стійкими до пасивних атак тоді і тільки тоді, коли несанкціонований користувач не має можливості відрізнити пусті контейнери від заповнених, зокрема методами візуального та статистичного аналізу.

Велика частина поширених програмних продуктів для прихованої передачі інформації методами комп'ютерної стеганографії, реалізують різні модифікації методу найменшого значущого біту, суть якого полягає у заміні молодших бітів контейнера бітами прихованого повідомлення. Користувач обирає довільний контейнер, розміри якого дозволяють розмістити у ньому повідомлення, і в результаті отримує стеганоконтейнер, що візуально не відрізняється від пустого [5]. Між молодшими бітами сусідніх елементів природних контейнерів, а також між молодшим та іншими бітами елементів контейнера існує кореляційний зв'язок, що може бути порушеним вкрапленням повідомлення. У цьому випадку для виявлення стеганоконтейнеру достатньо найпростішого аналізу – візуального аналізу бітових зрізів. Як правило, через наявність похибки квантування при оцифровці та інших шумів цифрові контейнери, що отримані з аналогових, більш стійкі до такої атаки, ніж ті, що були створені відразу цифровими. Разом з тим, вкрапливши повідомлення в НЗБ зашумленого контейнера, необхідно розподіляти його по всьому об'єму молодших бітів, інакше різниця між не зміненою та зміненою вкрапленням частинами може бути виявлена візуальною атакою на відповідний бітовий зріз.

Ємність. Ємність визначається як максимальна кількість даних повідомлення, які можуть бути вкраплennими в один елемент контейнера з дотриманням вимог невідчутності та стійкості.

Існують різні, іноді діаметрально протилежні підходи до визначення кількості приховуваної інформації на сьогоднішній день. Ці розбіжності обумовлені відмінностями в цілях захисту інформації, видах порушника, їх можливостях, типах контейнерів та повідомлень та іншими факторами. Зокрема, в якості теоретично досяжних границь, що не залежать від особливостей практичного застосування, використовують оцінку пропускної здатності, отриману в теоретико-інформаційній моделі стеганосистеми [1].

Ємність визначає потенційний об'єм інформації, яку можна приховати тим чи іншим методом стеганографії. А той об'єм, що був реально використаний в процесі стеганоперетворення певного контейнера (тобто вкраплення в нього додаткової інформації), будемо називати наповненістю контейнера. Очевидно, що в рамках тієї чи іншої стеганосистеми наповненість будь-якого контейнера не може перевищувати пропускної здатності створюваного нею стеганоканалу. Наповненість зручно вимірювати у відсотках від пропускної здатності. Так, наповненість заповненого контейнера складає 100%, порожнього – 0%.

1.3 Способи приховування даних

Багато змін трапилося з вічнозеленими носіями завдяки застосуванню стеганографії в комп'ютерних технологіях. Ці носії можуть бути віднесені до багатьох видів даних, таких як текст, диск, аудіо, зображення, звук, мережевий трафік або інші дані цифрової передачі даних. Способи приховування даних наведено нижче [8].

Приховування в тексті. Для приховування інформації у тексті (лінгвістична стеганографія) використовується звичайна надлишковість письмової мови або формати представлення тексту.

Найскладнішим об'єктом для приховування є електронна версія тексту, тому що його друкована версія може бути зображенням в електронному вигляді,

обробленим відповідними методами. Ця складність в основному обумовлена відносним дефіцитом у тексті надлишковості, на відміну від зображення або аудіо- файлу. В той час як існує можливість внести некритичні для ока модифікації у зображення або не відчутні для слухової системи людини (ССЛ) зміни у звучанні аудіофайлу, будь-яка зайва літера, зайвий символ або зайвий знак пунктуації може бути виявлений випадковим читачем [3].

Існують три основні методи приховування даних у тексті, що найширше розповсюджені:

– методи довільних інтервалів;

– синтаксичні методи;

– семантичні методи.

Приховування в зображеннях. В більшості випадків використовуються стеганографічні методи із графічними зображеннями в ролі контейнерів саме через такі причини:

– розповсюдження цифрових фотографій та відео, які необхідно захищати від протизаконного тиражування та розповсюдження;

– відносно великий об'єм графічних зображень, що дає широкий простір для приховування даних (великого розміру);

– розмір контейнера відомий заздалегідь, що дає змогу обирати оптимальний контейнер;

– відносно слабка чутливість людського ока до незначних змін у цифрових графічних зображеннях;

– добре розроблені, в останній час, методи цифрової обробки зображень.

Приховування у відео-файлах. Стеганографічні методи приховування рідше за все використовуються у відеоданих, оскільки даний файл складається з динамічних зображень (фреймів) та звукової доріжки. Варто також зазначити, що досі не використовуються як контейнери одночасно аудіодоріжки та фрейми.

На сьогодні існує три методи для приховування інформації у відеоданих, а саме:

Метод вбудовування на рівні коефіцієнтів – біти прихованого

повідомлення вбудовуються в коефіцієнти ДКП. Враховуючи, що використовуються алгоритми стиснення, основною проблемою стає накопичення зсуву та помилок. Для зменшення внесених змін використовують додатковий спеціальний сигнал. В зв'язку з обмеженням бітової швидкості при вбудовуванні змінюється лише 10-12% коефіцієнтів ДКП. При використанні даного методу прихована інформація зберігається при фільтруванні, зашумленні (адитивним шумом) і дискретизації [6].

Метод вбудовування інформації на рівні бітової площини – відрізняється високою пропускною здатністю і легкими обчисленнями. Але є й істотний недолік: інформація, вбудована таким чином, може бути легко видалена. При повторному накладенні послідовності біт якість відео погіршиться, а прихована інформація буде знищена.

Метод вбудовування інформації за рахунок енергетичної різниці між коефіцієнтами – в основі лежить диференціальне вбудовування енергії. Цей метод може використовуватись для багатьох алгоритмів стиснення. Інформація вбудовується шляхом видалення декількох коефіцієнтів ДКП [5].

В основному, приховування в відео використовує методи, які використовуються для приховування звуку та зображення, оскільки вже є відеозображеннями зображень і звуків. Відео складається з переміщення зображень в супроводі з аудіо. Насправді це є перевагою, оскільки будь-які невеликі спотворення користувачі навіть не помітять через неперервну кількість даних.

Приховування в аудіо-файлах. Особливий розвиток отримали стеганографічні методи приховування інформації у аудіосередовищі. Це охарактеризовано тим, що ССЛ працює у надширокому динамічному діапазоні і має доволі малий різницевий діапазон. Виходячи із цього, можна зробити висновок, що у аудіофайлах присутній широкий простір для приховування даних. Також ССЛ не здатна розрізняти абсолютну фазу, вирізняє лише відносну. Крім того, існують деякі види спотворень, викликаних зовнішнім середовищем, які можна використати для приховування даних [8]. Приховування даних в аудіо

файлах особливо складне через його великий діапазон частот. Аудіосигнали також чутливі до випадкових шумів. Шум може бути виявлений, якщо він знаходиться в діапазоні від одного до мільйону у звукових файлах. При приховуванні звуку користувач повинен скористатися перевагами слабкості людського слухового апарату, але також слід подбати про його високу чутливість.

1.4 Аналіз структури стеганографічної системи

Задачу вбудовування і виділення повідомлення з контейнера виконує стеганографічна система (стегосистема) [2]. Стегосистема складається з основних елементів, показаних на рис. 1.2.

Вбудовування повідомлень в контейнер проходить з використанням спеціального стегоключа. Ключ – псевдовипадкова послідовність біт, яку створює генератор, що задовольняє певним вимогам (криптографічно безпечний генератор). Цей ключ визначає порядок вбудовування повідомлення в контейнер. В якості основи генератора може використовуватися, наприклад, лінійний рекурентний регістр. Тоді адресатам для забезпечення зв'язку може повідомлятися початкове значення цього регістра [3].

Таємна інформація вбудовується у відповідності до ключа в ті відліки, спотворення яких не призводить до суттєвих спотворень контейнера. Ці біти утворюють стегошлях. Під суттєвим спотворенням можна розуміти спотворення, яке призводить як до неприйнятності для людини-адресата заповненого контейнера, так і до можливості виявлення факту наявності таємного повідомлення після стегоаналізу.

1.5 Огляд методів стеганографії

На даний час існує декілька способів приховування інформації або повідомлень у аудіо таким чином що зміни, внесені до аудіо файлу, є перцептивно нерозбірливими. Далі розглянуто загальні підходи [2,3].

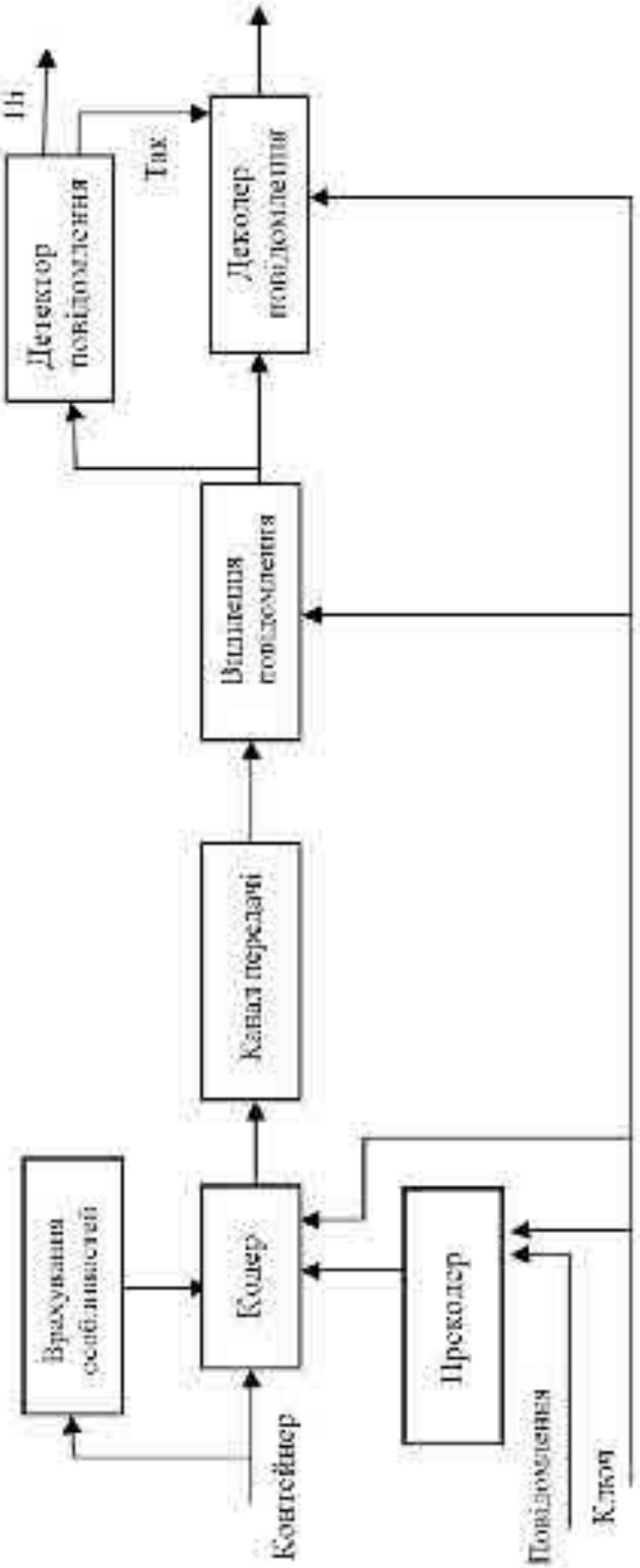


Рисунок 1.2. Структурна схема типової стего системи

1.5.1 Кодування за алгоритмом LSB

На сьогодні дуже популярною методологією є LSB (найменший значущий біт) алгоритм, який замінює найменш значущий біт в деяких байтах файлу обкладинки, щоб приховати послідовність байтів, що містять приховані дані. Це, як правило, ефективна методика у випадках, коли заміни LSB не викликають значне погіршення якості. У обчисленні, найменш значний біт (LSB) – це бінна позиція у двійковому цільовому числу, що дає одичні значення, тобто визначає чи є число парним або непарним. Іноді згадується LSB як найправильніший біт, завдяки конвенції в позиційному позначенні писати менш значущі цифри далі вправо. Він аналогічний найменш значущому знаку десяткового цілого числа, тобто цифра у крайній (праворуч) позиції. Бінарне представлення десяткового числа 149 з підсвічуванням LSB. MSB у 8-біговому двійковому значенні представляє значення 128 десяткових знаків. LSB представляє значення 1. Наприклад, щоб приховати букву "a" (ASCII-код 97, тобто 01100001) всередині восьми байт кришки, ви можете встановити LSB кожного байту таким чином:

10010010
01010011
10011011
11010010
10001010
00000010
01110010
00101011

Програма декодування зчитує вісім найменших значущих бітів з цих байтів, щоб відтворити прихований байт – це 0110001 – буква "a". Як можна зрозуміти, використовуючи цей метод, можна приховати байт у кожні вісім байт обкладинки. Існує п'ятдесят відсотків шансів, що біт, який замінюється, той самий, що і його заміна, тобто половину часу біт не змінюється, що допомагає мінімізувати якісну деградацію. Цей метод є одним з найпопулярніших, що вивчаються при приховуванні інформації цифрового звуку (а також інших типів носіїв). У цій техніці LSB, послідовності кожного зразка оцифрованого аудіофайлу замінюється на двійковий еквівалент секретного повідомлення. Це

найпростіший спосіб вставляти інформацію в цифровий аудіо файл. Це дозволяє приховати велику кількість даних у аудіофайлі або забезпечити високу швидкість вкладання без погіршення якості звукового файла [9].

Використання тільки одного LSB зразка аудіо-хосту дає потужність, еквівалентну частоті дискретизації, яка може варіюватися від 8 кбіт/с до 44,1 кбіт/с [2]. У LSB кодування ідеальна швидкість передачі даних становить 1 кбіт/с на 1 кГц. Однак у деяких варіантах кодування LSB, два найменш значущих біти замінюються двома бітами повідомлення. Це збільшує кількість даних, які можна кодувати, але також збільшує кількість шуму, що виникає, у аудіофайлі. Таким чином, слід враховувати вміст сигналу, перш ніж приймати рішення про використання операції LSB. Наприклад, звуковий файл, який був записаний на шумній станції метро, маскує низькошвидкісний шум кодування. З іншого боку, той же звук буде звучати в звуковому файлі, що містить фортепіано соло.

Щоб витягти секретне повідомлення з звукового файлу, закодованого LSB, одержувач потребує доступу до послідовності індексів вибірки, які використовуються в процесі вбудовування. Як правило, довжина секретного повідомлення, що підлягає кодуванню, менша, ніж зразки звукового файлу. Потім треба вирішити, як вибрати 7 підмножин зразків, які містять секретне повідомлення, і повідомити про це рішення одержувачу. Одна тривіальна методика полягає в тому, щоб почати з початку звукового файлу та виконувати кодування LSB, поки повідомлення не буде повністю вбудовано; залишаючи залишкові зразки незмінними. Це створює проблему безпеки, однак в тому, що перша частина звукового файлу матиме різні статистичні властивості, ніж друга частина не зміненого звукового файлу [2].

Одне рішення цієї проблеми полягає в тому, щоб поставити секретне повідомлення з випадковими бітами, так щоб довжина повідомлення дорівнювала загальній кількості зразків. Проте зараз процес введення закінчується зміною набагато більшої кількості зразків, ніж передача потрібного секрету. Це збільшує ймовірність того, що майбутній зломисник викриє секретне спілкування. Після виконання програми процедура може бути показана за допомогою оригінальних

даних, як показано на рис. 1.3. Більш витончений підхід полягає у використанні генератора псевдовипадкових чисел, щоб розповсюджувати повідомлення над звуковим файлом у випадковому порядку.

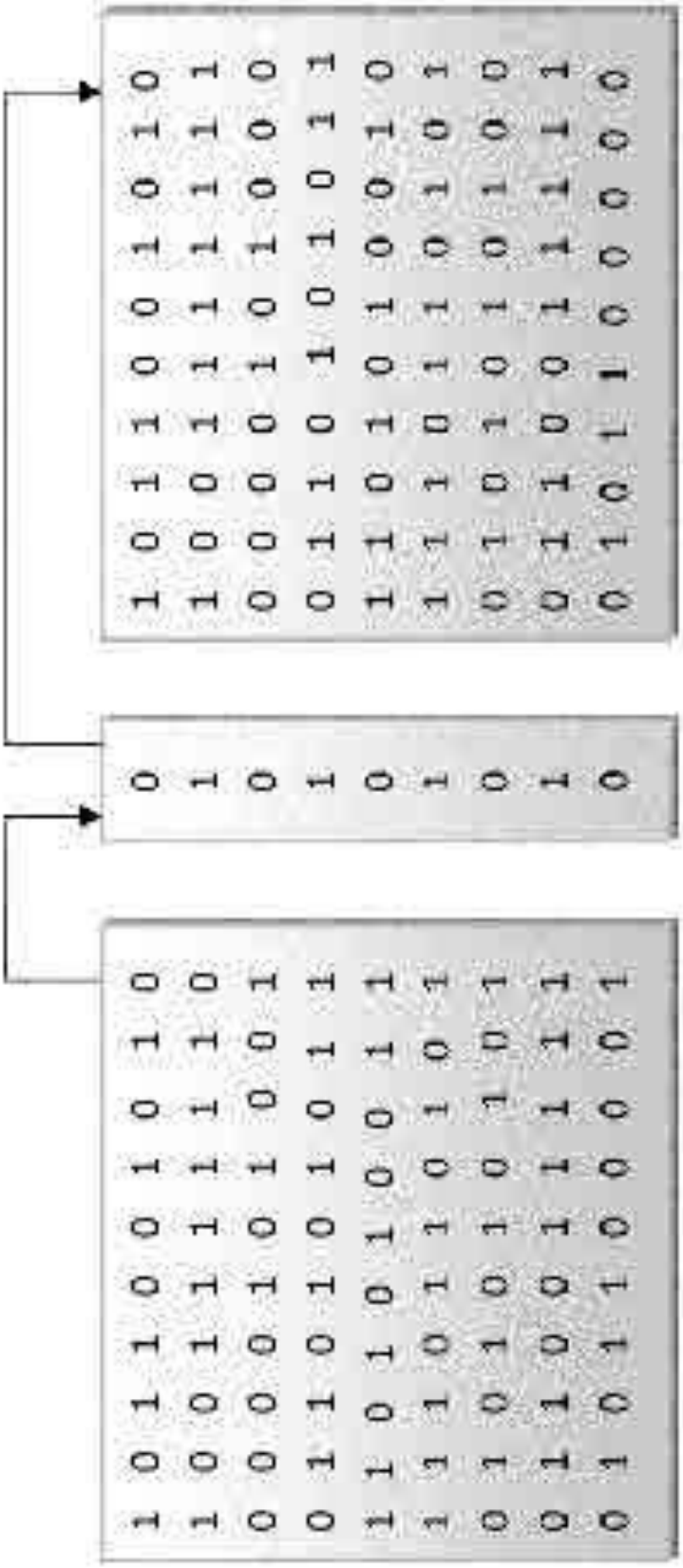


Рисунок 1.3. Метод LSB для аудіо стеганофонії

Одним із популярних підходів є використання методу випадкових інтервалів, в якому секретний ключ, яким володіє відправник, використовується як насіння в генераторі псевдовипадкових чисел для створення випадкової послідовності індексів вибірки. Приймач також має доступ до секретного ключа та знань генератора псевдовипадкових чисел, що дозволяє відновлювати випадкову послідовність показників вибірки. Однак перевірки повинні бути встановлені, щоб запобігти генерації псевдовипадкового числа двічі. Якщо це сталося, виникне зіткнення, коли зразок, уже змінений частиною повідомлення, буде змінено знову.

Проблему зіткнень можна подолати, відстежувати всі вже використані зразки. Інший підхід полягає у розрахунку підмножини зразків за допомогою псевдовипадкової перестановки всього набору за допомогою безпечної хеш-функції. Ця методика гарантує, що один і той же індекс ніколи не генерується більше одного разу [6].

1.5.2 Метод кодування парності

Кодування парності (паритетне кодування) – це один з надійних звукових стеганографічних методів. Замість того, щоб розбити сигнал на окремі зразки, цей метод розбиває сигнал на окремі зразки і вставляє кожен біт секретного повідомлення в біт парності. Якщо біт парності обраної області не збігається з секретним бітом, який буде кодуватися, процес інвертує LSB одного з зразків у регіоні. Отже, відправник має більше вибору при кодуванні секретного біту [3]. Використовуючи метод паритетного кодування, перші три біти повідомлення "НЕУ" закодовані на рисунку 1.4.

Декодування витягує таємне повідомлення, обчислюючи і виділяючи біти парності регіонів, що використовуються в процесі кодування. Знову ж таки, відправник і одержувач можуть використовувати загальний секретний ключ як насіння у генераторі псевдовипадкових чисел, щоб створити той самий набір зразків областей. Існує два основних недоліки, пов'язані з використанням таких методів, як кодування LSB або кодування різності.

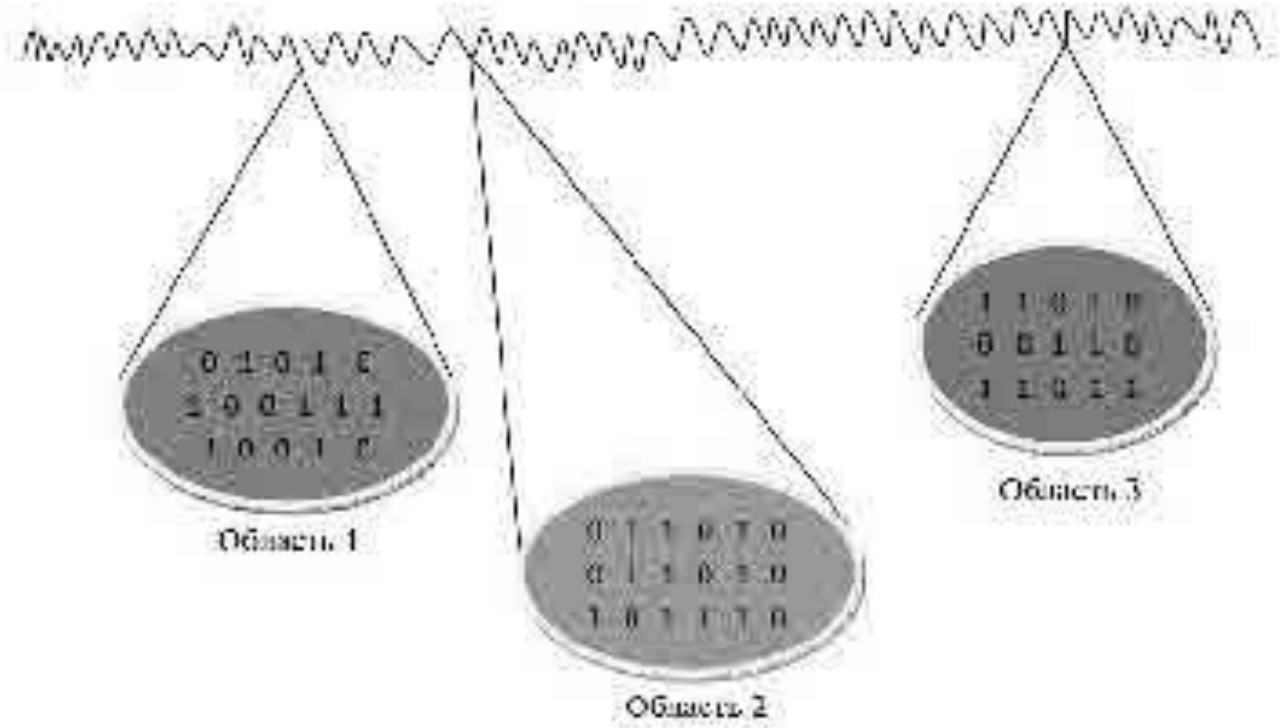


Рисунок 1.4. Метод кодування парності (паритетного кодування)

Людське вухо дуже чутливе і може часто виявляти навіть найменший шум, введений у звуковий файл, хоча метод паритетного кодування досить наближений до того, щоб введений шум був не чутним. Обидва способи мають ще один недолік, оскільки вони не є надійними. Якщо звуковий файл, вбудований в секретне повідомлення з кодуванням LSB або кодування різності, буде пошкоджений зразком, вбудована інформація буде втрачена [6]. Потужність може дещо покращити, використовуючи техніку резервування при кодуванні секретного повідомлення. Однак технології резервування значно зменшують швидкість передачі даних.

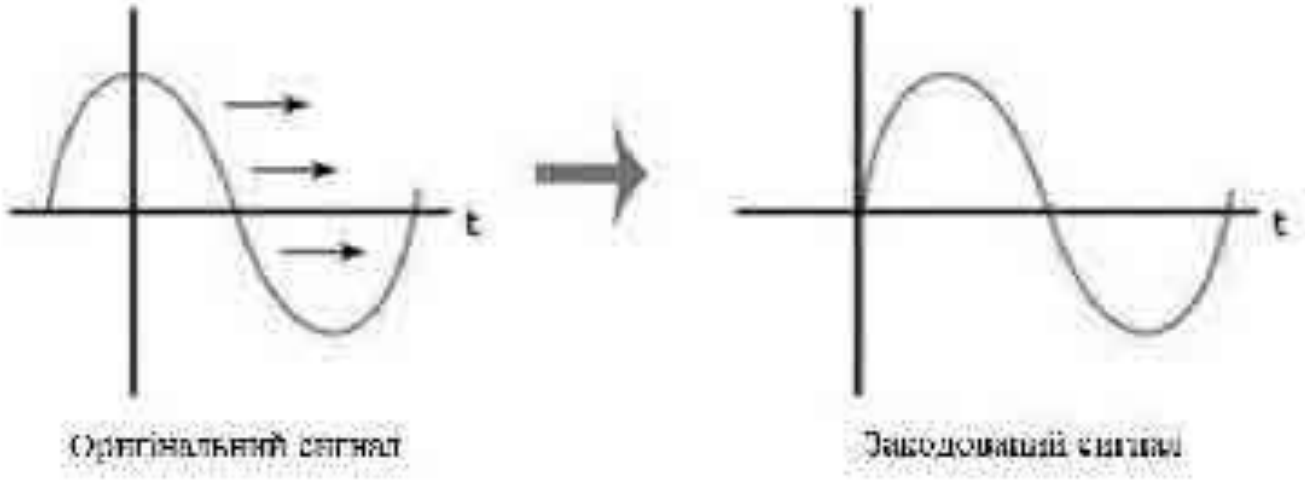
1.5.3 Метод кодування фази

Технологія фазового кодування працює шляхом заміни фази початкового аудіо сегменту з еталонною фазою, що представляє секретну інформацію. Решта фази сегментів коригується для збереження відносної фази між сегментами. З точки зору співвідношення сигнал / шум, фазове кодування є одним з найбільш ефективних методів кодування. Коли відбувається різка зміна фазового зв'язку між кожною частотною складовою, спостерігається помітна дисперсія фази.

Однак до тих пір, поки модифікація фази буде достатньо мала, може бути досягнуте неголосне кодування [5]. Цей метод спирається на те, що фазові компоненти звуку не так сприймаються людському вухо, як це звучить. Фазове кодування розглядає недоліки шумопоглинаючих методів аудіо-стеганографії. Фазове кодування залежить від того, що фазові компоненти звуку не настільки чутливі для людського вуха, як шум. Замість того, щоб вводити збурення, ця техніка кодує біти повідомлень у вигляді фазових зрушень у фазовому спектрі цифрового сигналу, досягаючи безшумного кодування у співвідношенні сигнал/шум. Або можна сказати, що фазове кодування залежить від заміни вибраних фазових компонентів прихованими даними. Відзначено, що серед усіх методів прихокування, фазове кодування перешкоджає кращому перекручуванню сигналу. Фазове кодування встановлює дані в фазові компоненти, використовуючи незалежну багатодіапазонну фазову модуляцію. У такому підході непомітна фазова модифікація досягається за допомогою керованої фазової зміни аудіо-хосту, показаного на рисунку 2б. Оригінальний звуковий сигнал розбитий на менші сегменти, довжини яких дорівнюють розміру кодованого повідомлення.

Кодування фази пояснюється в наступному порядку:

- Розділіть оригінальний звуковий сигнал на менші сегменти, такі, щоб довжини були такого ж розміру, як і розмір кодованого повідомлення;
- Матриця фаз створюється шляхом застосування дискретного перетворення Фур'є (DFT);



Оригінальний сигнал Закодований сигнал

Рисунок 1.5. Метод фазового кодування

– Обчислити відмінності фаз між суміжними сегментами;

– Фазові зрушення між суміжними сегментами легко виявляються. Це означає, що ми можемо змінити абсолютні фази сегментів, але відносні відмінності фаз між суміжними сегментами повинні бути збережені. Таким чином, секретна інформація встановлюється тільки в вектор фази першого сегмента сигналу наступним чином:

$$фаза = \begin{cases} \frac{\pi}{2}, якщо\ біт = 0 \\ -\frac{\pi}{2}, якщо\ біт = 1 \end{cases} \quad (1.1)$$

– Використовуючи нову фазу першого сегмента, створюється нова фазова матриця та вихідні відмінності фаз;

– Звуковий сигнал реконструюється шляхом застосування зворотного дискретного перетворення Фур'є з використанням нової фази матриці та матриці оригінальної величини, а потім об'єднує сегменти звуку назад.

Для вилучення секретної інформації з звукового файлу приймач повинен знати довжину сегмента. Тоді приймач може використовувати ДПФ, щоб отримати етапи та витягнути секретну. Одним з недоліків, пов'язаних з фазовим кодуванням, є низька швидкість передачі даних через те, що таємне повідомлення кодується лише в першому сегменті сигналу. Це може бути вирішено шляхом збільшення довжини сегмента сигналу [7]. Однак це змінить фазові зв'язки між кожною частотною складовою сегмента більш різко, що робить кодування легшим для виявлення. Як наслідок, метод фазового кодування використовується, коли потрібно приховати лише невелику кількість даних, таких як водяний знак.

1.5.4 Метод розповсюдження спектру

Основний метод розповсюдження спектру у аудіо-стеганографії намагається поширювати секретну інформацію по частотному спектру аудіо-сигналу. Це схоже на систему, яка використовує реалізацію LSB, яка поширює біти повідомлень випадковим чином по всьому звуковому файлу. Проте, на

відміну від кодування LSB, метод розповсюдження спектру поширює секретну інформацію по частотному спектру звукового файлу за допомогою коду, який не залежить від фактичного сигналу [2]. Як результат, кінцевий сигнал займає смугу пропускання, яка перевищує те, що дійсно потрібно для передачі. Метод розповсюдження спектру здатний сприяти поліпшенню продуктивності в деяких областях порівняно з кодуванням LSB та фазовим кодуванням, оскільки він забезпечує помірну швидкість передачі даних і високий рівень надійності відносно методів видалення. Проте метод розповсюдження спектру має один основний недолік, який може вводити шум у звуковий файл.

1.5.5 Метод приховування відлуння

Метод приховування відлуння використовує секретну інформацію у звуковому файлі, вводячи відлуння в дискретний сигнал. Приховування відлуння має переваги забезпечення високої швидкості передачі даних і вищої надійності в порівнянні з іншими методами. Лише один біт секретної інформації може бути закодований, якщо тільки вихідний сигнал було отримав лише одне відлуння.

Отже, перед початком процесу кодування оригінальний сигнал розбитий на блоки. Після завершення процесу кодування блоки об'єднуються разом, щоб створити остаточний сигнал [6]. Для успішного приховання даних, три параметри відлуння повинні бути різними: амплітуда, швидкість розпаду та зміщення (час затримки) від вихідного сигналу. Всі три параметри встановлені нижче порогу слуху людини, тому відлуння не може бути легко виявлено. Крім того, зміщення змінюється, щоб представляти бінарне повідомлення для кодування. Одне значення зміщення являє собою двійкове значення, а значення другого зміщення являє собою двійковий нуль. Зміщення представлено на рисунку 1.6.

Кодування може містити лише один біт інформації, якщо вихідний сигнал був вироблений лише одним відлунням. Тому початковий сигнал розбивається на блоки, перш ніж процес кодування починається. Після завершення процесу кодування, блоки об'єднуються разом, щоб створити остаточний сигнал [9].

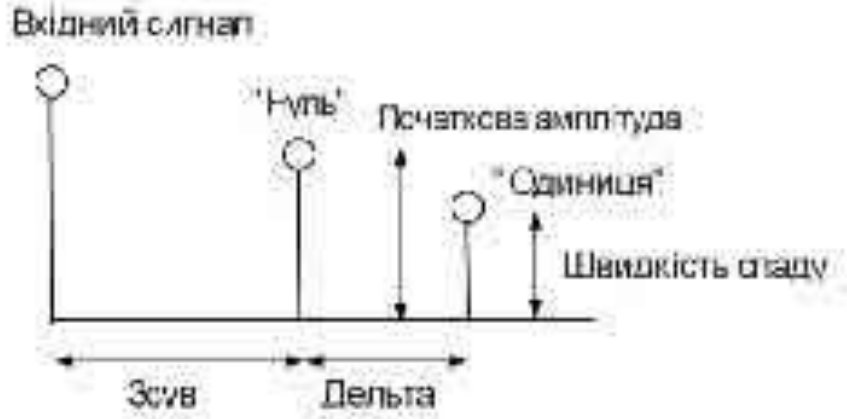


Рисунок 1.6. Значення відступів відтворюють двійкові значення

Тепер ми пройдемо просту форму процесу приховування відлуння, використовуючи повідомлення "HEY". Для стислості сигнал буде повністю розділений на блоки, хоча за звичайних умов випадкова кількість зразків між кожною парою блоків повинна залишатись невикористаною, щоб зменшити ймовірність виявлення. Перестановка остаточних блоків представлена на рисунку 1.7.

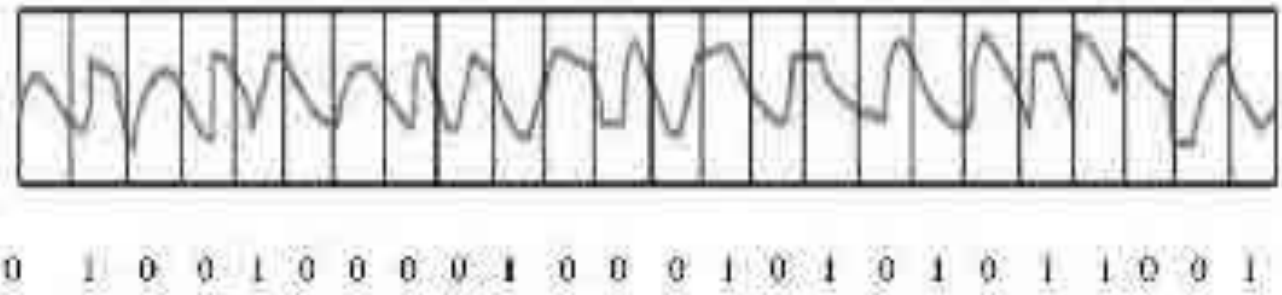


Рисунок 1.7. Блоки переставляються для отримання остаточного сигналу

Сигнал ділиться на блоки, і кожному блоку присвоюється один чи нуль, що базується на секретному повідомленні. У цьому випадку повідомлення є двійковим еквівалентом "HEY". Використовуючи цю реалізацію процесу приховування відлуння, зазвичай може бути отриманий сигнал, який має досить помітний набір відлуння, що підвищує ризик виявлення. Друга реалізація процесу приховування відлуння вирішує цю проблему. Спочатку відлуння створюється з усього вихідного сигналу, використовуючи значення двосмугового зсуву нуля. Потім другий сигнал відлуння створюється з усього вихідного сигналу, використовуючи значення двосмугового зсуву. Таким чином, "один" відлуння

містить тільки одиниці, а "нульовий" відлуння містить тільки нулі. Щоб об'єднати два відлуння разом, щоб отримати остаточне кодування, використовуються два сигнали змішувача. Сигнали змішувача мають значення як одиниці, так і нулі, в залежності від того, який біт потрібно кодувати в блоці. "Один" сигналу відлуння помножується на "один" сигнал змішувача, а "нульовий" сигнал відлуння помножується на "нульовий" сигнал змішувача. Потім два результати додаються разом, щоб отримати остаточний сигнал. Остаточний сигнал є менш різким, ніж той, який був отриманий за допомогою першої реалізації приховування відлуння. Це пояснюється тим, що два ефекти змішувача є доповненнями один до одного, і ці переходи використовуються в кожному сигналі. Ці дві характеристики сигналів змішувача забезпечують більш плавні переходи між відлуннями. Щоб витягти таємне повідомлення зі стего-сигналу, приймач повинен мати можливість розбити сигнал на той же блоковий порядок, який використовується у процесі кодування сигналу.

1.5.6 Порівняння методів стеганофонії

Визначимо недоліки попередньої процедури та те, як вони відрізняються від поточного методу. Основні недоліки, пов'язані з використанням існуючих методів, таких як приховування відлуння, розповсюдження спектру та паритетне кодування, є дуже чутливим до шуму, і вони часто можуть вилякати навіть найменший шум, введений у звуковий файл, і інша проблема – це надійність. Фазове кодування має основний недолік низької швидкості передачі даних через те, що секретне повідомлення кодується тільки в першому сегменті сигналу. Отже, цей метод використовується лише тоді, коли потрібно передати невелику кількість даних. Серед різних методів приховування інформації, запропонованих для вбудовування секретної інформації в аудіофайл, найменш значущий біт (LSB) є найпростішим способом вбудовування секретної інформації в цифровий аудіо файл, замінивши найменш значущий біт аудіофайла з двійкового повідомлення. Тому метод LSB дозволяє кодувати велику кількість секретної інформації у аудіофайлі. Порядок приховування секретної інформації за допомогою LSB:

– Приховати аудіо файлу бітовий потік.

– Перетворення кожного символу секретної інформації в бітовий потік.

– Заміна біту звуку LSB на біт символу LSB у секретній інформації.

1.6 Вибір методу стеганофонії

У межах підрозділу розкрито теоретичну сторону існуючих алгоритмів розв'язання задачі аудіо-стеганографії. Проведено порівняння та розглянуто переваги та недоліки описаних методів. Виконано аналіз структури стеганографічної системи.

Метод LSB забезпечує більшу безпеку та є ефективним способом приховування секретної інформації від хакерів і відправлення в пункт призначення безпечним та невиявленим способом. Ця запропонована система також гарантує, що розмір файлу не змінюється навіть після кодування, і також підходить для будь-якого типу формату аудіофайлів. Також він дозволяє приховувати в файлах контейнерах набагато більший об'єм секретної інформації в порівнянні з іншими алгоритмами. Через його переваги над іншими алгоритмами, він був обраний для розробленння модифікації, яка описана в наступному підрозділі.

1.7 Необхідність модифікації алгоритму LSB

Основні недоліки використання таких стеганофонічних методів як відлуння, розширеного спектру і паритетного кодування полягають в тому, що вони вносять шум в аудіо файл, який може бути досить помітним для людського вуха, а також надійність даних методів викликає питання. Щодо фазового кодування, то цей метод має основний недолік, що полягає в низькій швидкості передачі даних через те, що секретне повідомленння кодується тільки на першому сегменті сигналу. Отже, цей метод використовується тільки тоді, коли передається невелика кількість даних. Метод найменшого значущого біта (LSB) є найпростішим методом для вбудовування секретної інформації серед запропонованих вище методів стеганографії. Метод LSB дозволяє закодувати велику кількість даних в звуковий файл, забезпечує більш високий рівень безпеки

в порівнянні з іншими методами, є ефективним методом для приховування секретної інформації від зловмисників, а також гарантує незмінність розміру файлу навіть після кодування і підходить для будь-якого типу формату аудіо файлу. Також він дозволяє приховувати в файлах контейнерах набагато більший об'єм секретної інформації в порівнянні з іншими алгоритмами. Таким чином, проблема полягає в помінному спотворенні вхідного контейнеру та відсутності стійкості до атак.

1.8 Складання математичної моделі

Етапи приховування повідомлення можуть бути представлені так:

$$E: C \times M \rightarrow S, \qquad (1.2)$$

де $S = \{ (c_1, m_1), (c_2, m_2), \dots , (c_n, m_n), \} = \{s_1, s_2, \dots , s_n\}$ – множина заповнених контейнерів (стегано-контейнерів), E – відображення, C – представляє всі можливі файли, в які будуть приховані секретні дані, а M – всі можливі секретні повідомлення.

Для вилучення будь-якого таємного повідомлення з файлу, який його містить, використовується наступне:

$$D: M \times C \rightarrow M \qquad (1.3)$$

з необхідною умовою – відсутністю перетину, тобто якщо $m_a \neq m_b$, при чому $m_a, m_b \in M$ та $(c_a, m_a), (c_b, m_b) \in S$, то $E(c_a, m_a) \cap E(c_b, m_b) = \emptyset$

В загальному випадку стеганосистему можна представити як сукупність $\sum(C, M, S, E, D)$ – контейнерів, повідомлень та перетворень, що їх зв'язують. Завжди контейнери C обираються таким чином, щоб заповнений контейнер майже не відрізнявся від порожнього контейнера.

Стеганосистема може вважатися надійною, коли:

$$sim[c, E(c, m)] = 1 \qquad (1.4)$$

де sim – функція подібності.

Сам контейнер може обиратися двома способами: довільно (сурогатний метод) та підбором найбільш придатного у конкретному випадку контейнера,

який зміниться найменше при перетворенні. В останньому випадку контейнер обирається виходячи із умови:

$$c = \max sim\,[c, E(c, m)] \qquad (1.5)$$

У будь-якому випадку пряме та зворотне перетворення (E та D) мають відповідати одне одному та підлягати умові, що незначне викривлення контейнера (на величину δ) не має призводити до викривлення прихованої інформації:

$$E\,(c, m) \approx E\,(c + \delta, m) \qquad (1.6)$$

$$D\,[E(c, m)] \approx D\,[E(c + \delta, m)] = m \qquad (1.7)$$

1.9 Підбір вхідних даних

У даній роботі як файли контейнери, так і секретні повідомлення є MP3-файлами, оскільки вони забезпечують добре стиснення даних і є найбільш поширеними. І, враховуючи обмеження людського слуху, стиснення даних на бітрейтах біля 320 Кбіт/с майже не впливає на сприйняття якості звуку.

Взагалі, більшість дослідників використовують файли формату wav, що призводить до наявності стандартного набору даних для нього. На рис.1.8 показано варіант алгоритму стиснення wav-файлу із вбудовуванням секретного біту та подальшим формуванням MP3-кадру.

В нашому випадку, при використанні MP3-файлів, обрано власний набір всіх вхідних даних. У цьому наборі даних міститься 10 різних жанрів: Класична, Джаз, Кантрі, R&B, Реп, Реггі, Поп, Рок, Блюз, Хіп-хоп. Генерування MP3-файлів виконується програмою для перетворення з WAV-файлу у MP3-файл. Найбільш популярними є п'ять різних ступенів стиснення (бітрейтів) файлів MP3: 320 Кбіт/с, 256 Кбіт/с, 196 Кбіт/с, 128 Кбіт/с і 96 Кбіт/с. Їх значення відрізняються впливом на якість звуку. Іншими словами, збільшення кількості бітів на зразок призводить до підвищення якості звуку.

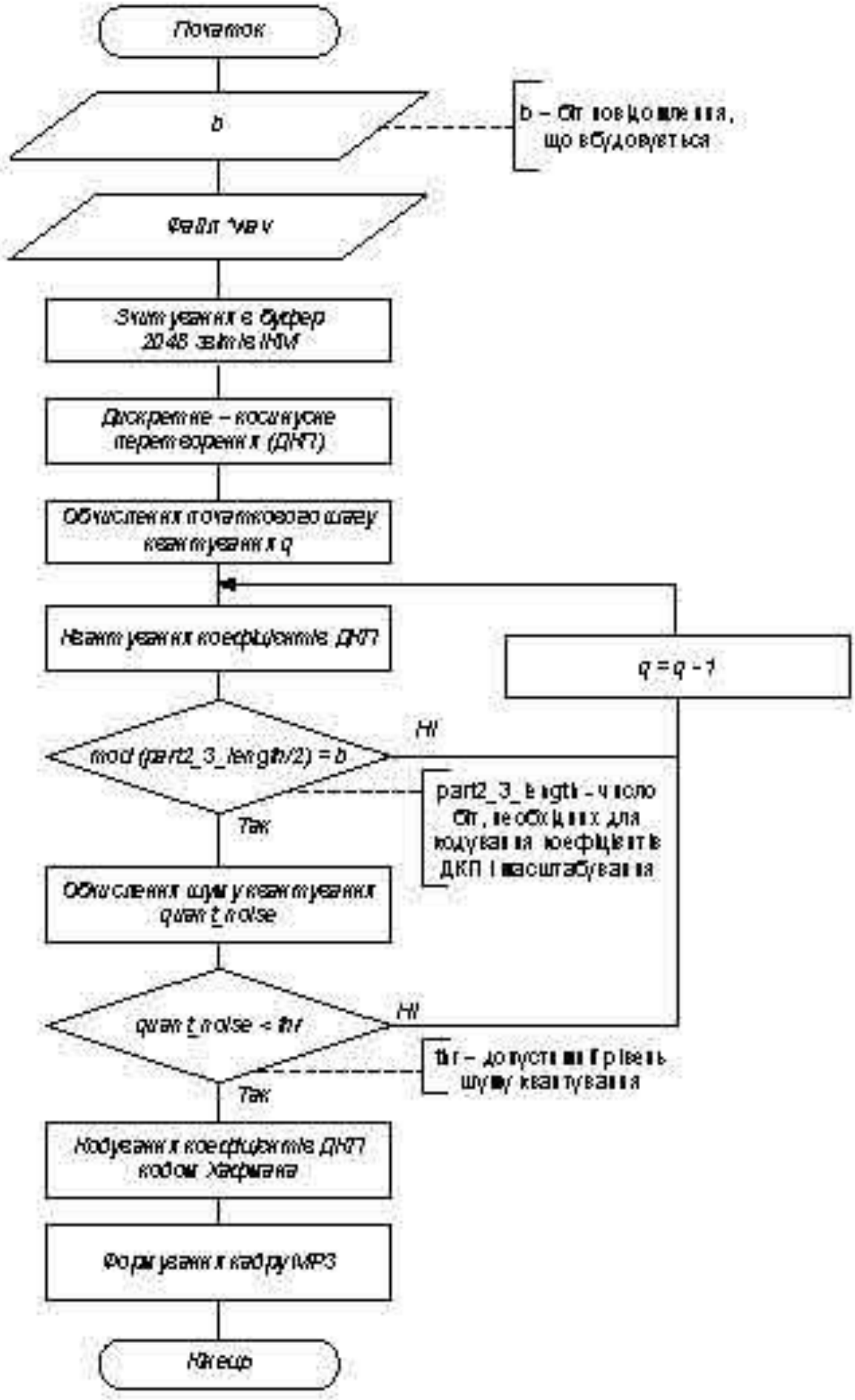


Рисунок 1.8. Алгоритм стиснення інформації

Наступна таблиця ілюструє стандартний набір даних MP3, який створений для використання у даній роботі.

Таблиця 1.1. Набір вхідних даних (використовуються як секретне повідомлення)

Найменування жанру	*Час, сек.*	*Розмір файлу (WAV), MB*	*Розмір файлу (320 кбіт/сек), MB*	*Розмір файлу (256 кбіт/сек), MB*	*Розмір файлу (192 кбіт/сек), MB*	*Розмір файлу (128 кбіт/сек), MB*	*Розмір файлу (96 кбіт/сек), MB*
Класична	2:42	14.4	6.54	5.24	3.94	2.62	1.97
Джаз	2:56	15.4	7.01	5.60	4.21	2.81	2.11
Канрі	3:11	16.5	7.51	6.00	4.51	3.01	2.26
R&B	3:15	16.9	7.68	6.15	4.62	3.08	2.32
Реп	3:24	17.4	7.90	6.33	4.76	3.17	2.38
Ретті	3:42	18.2	8.27	6.62	4.98	3.32	2.49
Поп	3:53	19.1	8.68	6.95	5.22	3.48	2.62
Рок	4:04	20.3	9.22	7.38	5.55	3.70	2.78
Блюз	4:12	21.1	9.59	7.67	5.77	3.85	2.89
Хіп-хоп	4:27	22.7	10.31	8.25	6.21	4.14	3.11

У даній роботі вхідними даними є аудіо-файли MP3 (у якості файлів контейнерів) – по 20 файлів кожного із вище згаданих стилів музики.

1.10 Контроль цілісності файлів після атаки

Значення хеш-функції. Це типовий метод перевірки цілісності даних, який широко використовується в різних протоколах та додатках. Він має суттєву роль у поточній криптографії. Основна ідея щодо хеш-функцій полягає в тому, що хеші виступають як компактний делегат, який називається відбитками чи цифровими відбитками вхідного об'єкту.

Контрольна сума. Контрольна сума є одним з основних методів перевірки цілісності. Її значення залежить від порівняння вхідного об'єкту зі значеннями, отриманими після кодування. Цей метод в основному застосовується разом з розрахунком значень хеш-функцій. Метод порівняння значень контрольної суми двох об'єктів допомагає виявити зміни цілісності. Однак він не може відновлювати дані через невідповідність між вхідними та вихідними значеннями контрольних сум. Збережені контрольні суми можуть бути пошкоджені або змінені. Ще однією причиною проблеми відновлення значення контрольної суми є те, що вона обчислюється за допомогою односторонньої хеш-функції, коли дані

не можна відновити, щоб отримати значення контрольної суми.

1.11 Виконання модифікації методу LSB

В даній роботі виконується модифікація існуючого методу аудіо-стеганографії найменш значущого біту (LSB). Було вирішено модифікувати саме цей метод, оскільки він надає змогу вкраплення набагато більших об'ємів повідомлень в порівнянні з іншими методами, але при цьому, сам метод можна вдосконалити з метою зменшення помітних змін файлів контейнерів. Розроблена модифікація полягає у чергуванні місця заміни не значущого біту серед найменш значущих кожного зразка в файлі-контейнері бітом секретного повідомлення для підвищення безпеки.

На протязі дослідження мали бути виконані основні етапи: підготовка контейнеру та секретного повідомлення, вкраплення секретного повідомлення, оцінка спотворення вхідного файлу, перевірка цілісності контейнерів, застосування атаки. Підготовка контейнеру та секретного повідомлення полягає у перевірці можливості вкраплення секретного повідомлення до контейнеру та перетворенні у двійковий формат даних. На етапі вкраплення секретного повідомлення здійснюється саме приховування секретного повідомлення до файлу-контейнеру наступними методами: традиційні 4-LSB, 2-LSB, 1-LSB та розроблена модифікація. Оцінка спотворення вхідного файлу перевіряється за допомогою розрахунку коефіцієнту пікового сигналу до шумового співвідношення для кожного з стегано-контейнерів. Перевірка цілісності контейнерів полягає у розрахунку значень контрольної суми стегано-контейнерів, знаходженні хеш-функції стегано-контейнерів та зміні частот стегано-контейнерів. Етап застосування атаки полягає в додаванні адитивного гаусового білого шуму до стегано-контейнерів з метою їх спотворення. Потім знову розраховуються показники цілісності та спотворення для порівняння із значеннями до застосування атаки. Атака здійснюється з метою перевірки ефективності розробленої модифікації алгоритму.

Для оцінки продуктивність розробленої модифікації методу, до стегано-контейнеру додають адитивний гаусовий білий шум (AWGN) з різними

значеннями дисперсії, перш ніж витягати таємне повідомлення. Після цього обчислюються значення співвідношення (PSNR) пікового сигналу до шумового та порівнюються з результатами, отриманими перед додаванням шуму.

1.11.1 Підготовка секретного повідомлення та контейнеру

У даному дослідженні вхідні дані (файли-контейнери та секретне повідомлення) є аудіо-файлами MP3. На цьому кроці контейнер спочатку перетворюється з десяткового формату даних у двійковий. Після підготовки контейнеру секретне повідомлення підготовлюється до процесу вбудовування у файл-контейнер. Значення звукового сигналу секретного повідомлення перетворюються в позитивні значення, а потім перетворюються з десяткового формату даних у двійковий. Після цього виконується етап перевірки, чи є довжина секретного повідомлення меншою, ніж довжина файлу-контейнеру. Якщо довжина більша – обчислення зупиняються.

Якщо аудіо-файл є моно-звуком – створюється вектор одного стовпця, а якщо стереозвуком – матриця подвійного стовпця (лівий і правий канали) і потім робота продовжується з середнім значенням цих стовпців. Далі виконується етап перевірки, чи є швидкість передачі даних (Кбіт/сек) секретного повідомлення меншою, ніж швидкість передачі даних файлу-контейнеру. Файл-контейнер має бути придатним для вкраплення секретного повідомлення у форматі розміру. Якщо файл контейнер не є придатним – розрахунок зупиняється, інакше – виконується наступний етап приховування даних.

1.11.2 Вбудовування секретного повідомлення у файл

При виконанні комплексного дослідження секретне повідомлення приховується 4 способами: традиційні 4-LSB, 2-LSB, 1-LSB та розроблена модифікація. Для традиційної техніки в кожному з ітерацій циклу вибраний байт змінюється за допомогою такої логіки:

– якщо використовується 4-LSB, то біти з 2-го до 5-го замінюються першими доступними 4 бітами у секретному повідомленні;

– якщо використовується 2-LSB, то 2-й і 3-й біти замінюються першими

доступними бітами у секретному повідомленні;

– якщо використовується 1-LSB, тільки другий біт замінюється першим бітом, доступним у секретному повідомленні.

Біти приховані наступним чином для розробленої модифікації у кожному з циклів ітерацій (рис.1.9): 1 біт приховується у першому байті; 2 біти приховується у другому байті; 2 біти приховується у третьому байті; 1 біт приховується у четвертому байті. Модифікований байт після цього перетворюється з двійкового на десятковий щоб створити стегано-об'єкт.

1.11.3 Аналіз спотворення вхідного файлу

На цьому етапі розраховується коефіцієнт пікового сигналу до шумового співвідношення (PSNR) та середня квадратична похибка (MSE). Обидва відображають дві метрики помилок, які використовуються для порівняння якості обкладинки. PSNR та MSE можна обчислити за допомогою наступних формул, відповідно:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(X(i) - Y(i))^2 \qquad (1.8)$$

$$PSNR = 10\lg\frac{(MAX)^2}{MSE} \qquad (1.9)$$

де X – оригінальний об'єкт, Y – це стего-об'єкт, N – розмір обкладинки, MAX – максимальне значення амплітуди вхідного аудіо-файлу.

1.11.4 Перевірка цілісності контейнерів

На даному етапі застосовуються різні методи обробки, коли кожен метод, кодова книга зберігається і порівнюється пізніше з процесом вилучення.

З метою порівняння стегано-контейнерів з вхідними файлами пропонується розрахунок наступних показників:

– значення контрольної суми стегано-контейнеру та вхідного файлу, потім розраховується значення подібності даних у відсотках;

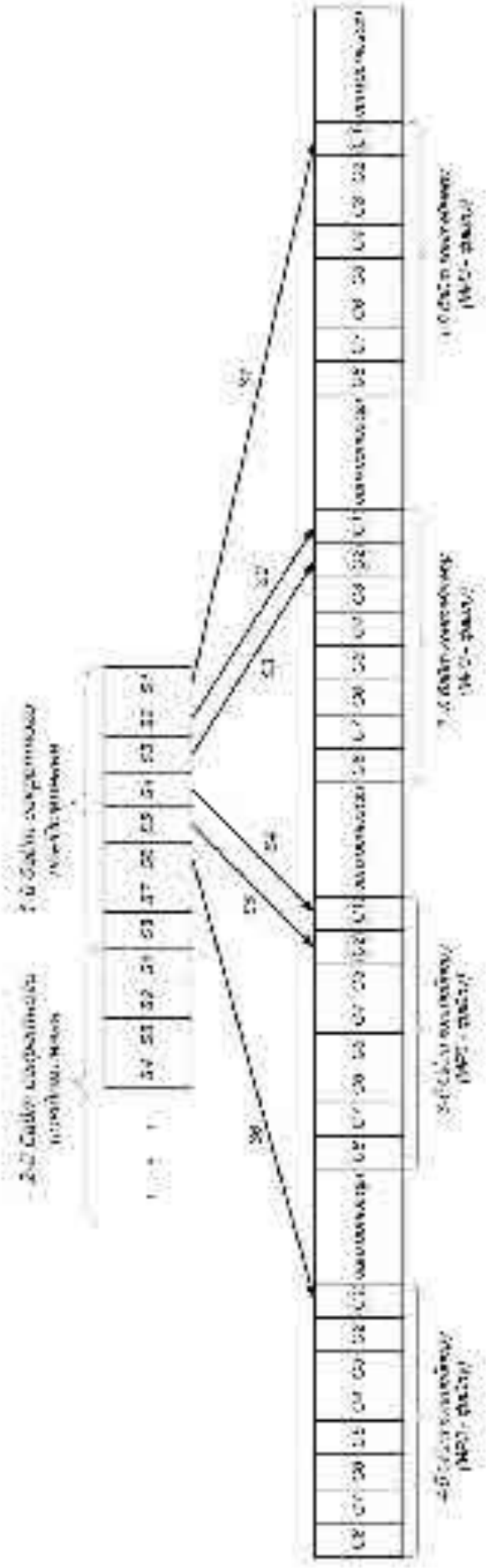


Рисунок 1.9. Модифікація методу LSB

– знаходження хеш-функцій стегано-контейнерів та вхідних даних, потім у відсотках розраховується, наскільки два файли подібні між собою;

– розраховується зміна частот у відсотках секретного повідомлення, витягнутого зі стегано-контейнера відносно вхідного секретного повідомлення.

1.11.5 Виконання атаки

Для оцінки продуктивності розробленого методу, перед етапом вилучення секретного повідомлення до стегано-контейнеру додають адитивний гаусовий білий шум (AWGN) з різними значеннями дисперсії і тоді значення пікового сигналу до шумового співвідношення (PSNR) обчислюються та порівнюються з результатами, отриманими до додавання цього шуму.

1.12 Опис засобів розробки для реалізації програмного продукту

Для створення програмного продукту аудіо-стеганографії були використані такі засоби для програмування на мові C#, як Microsoft Visual Studio 2019 та Windows Forms. Мова C# проста у використанні та водночас повноцінна мова програмування, що надає багато засобів для структурування і підтримки великих програм та рішень. Вона краще за C/C++ обробляє помилки, і, будучи мовою високого рівня, має вбудовані типи даних високого рівня, такі як гнучкі масиви, списки і словники, ефективна реалізація яких на мові C потребує значних витрат часу. Також для розширення функціональності можна використовувати готові бібліотеки, які отримуються напряму в середовища розробки через вбудований у Visual Studio 2019 менеджер пакетів NuGet Package Manager.

Мова програмування C# дозволяє розбивати програми на модулі, що потім можуть бути використані в інших програмах. C# поставляється з великою кількістю стандартних бібліотек, які можна використовувати, як основу для нових програм або як приклади при вивченні мови. Стандартні модулі надають засоби для роботи з файлами, системними викликами, мережними з'єднаннями і навіть інтерфейсами до різних графічних бібліотек. C# дозволяє писати зручні для читання програми завдяки загальноприйнятим узгодженням щодо написання

коду та назв полів різних типів. Програми, написані мовою C++ звичайно значно коротші ніж їхні еквіваленти на C з декількох причин:

– типи даних високого рівня дозволять виразити складні операції однією інструкцією;

– наявність новітніх методів;

– широкий вибір методів та структур.

Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, наслідування, перевантаження операторів, інкапсуляцію, закриття методів, вказівники на функції та члени класів, атрибути, події, властивості, делегати, винятки, коментарі у форматі XML. Перейнявши багато чого від своїх попередників (мов C++, Delphi і Smalltalk) – C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, множинне успадкування класів (на відміну від C++) [8].

Windows Forms дозволяє розробляти інтелектуальні клієнти. Інтелектуальний клієнт – це програма з повнофункціональним графічним інтерфейсом, просте в розгортанні і оновленні, здатне працювати при наявності або відсутності підключення до Інтернету і використовує більш безпечний доступ до ресурсів на локальному комп'ютері в порівнянні з традиційними додатками Windows. Windows Forms – це технологія інтелектуальних клієнтів для NET Framework. Вона являє собою набір керованих бібліотек, що спрощують виконання стандартних завдань, таких як читання з файлової системи і запис в неї При використанні середовища розробки, як Visual Studio, можна створювати інтелектуальні клієнтські програми Windows Forms, які відображають відомості, запитують введення від користувачів і обмінюються даними з віддаленими комп'ютерами по мережі. У Windows Forms, форма – це візуальна поверхня, на якій виводиться інформація для користувача. Зазвичай додаток Windows Forms будується шляхом приміщення елементів управління на форму і написання коду для реагування на дії користувача, такі як клацання миші або натискання клавіш.

Елемент управління – це окремий елемент призначеного для користувача інтерфейсу, призначений для відображення або введення даних. При виконанні користувачем якої-небудь дії з формою або з одним з елементів управління створюється подія. Додаток реагує на ці події за допомогою коду і обробляє події при їх виникненні. Windows Forms включає широкий набір елементів управління, які можна додавати на форми: текстові поля, кнопки, списки, що розкриваються, перемикачі та навіть веб-сторінки. Якщо існуючий елемент управління не задовольняє потребам, в Windows Forms можна створювати власні елементи управління. До складу Windows Forms входять багатофункціональні елементи призначені для користувача інтерфейсу, що дозволяють відтворювати можливості таких складних додатків, як Microsoft Office. Використовуючи необхідні елементи управління, можна створювати панелі інструментів і меню, що містять текст і малюнки, та інші елементи управління, такі як текстові поля і поля зі списками.

За допомогою Visual Studio можна легко створювати додатки Windows Forms. Досить вибрати елемент керування курсором і помістити його в потрібне місце на формі. Для подолання труднощів, пов'язаних з вирівнюванням елементів управління, конструктор надає такі додаткові елементи, як лінії сітки і лінії прив'язки. За допомогою Visual Studio або компіляції з командного рядка, можна використовувати елементи управління для створення складних макетів форм за менший час. У багатьох додатках потрібно відображати дані з бази даних, XML-файлу, веб-служби XML або іншого джерела даних. Windows Forms надає гнучкий елемент управління для відображення таких табличних даних в традиційному форматі рядків і стовпців так, що кожен фрагмент даних займає свою власну клітинку. За його допомогою можна, налаштувати зовнішній вигляд окремих осередків, зафіксувати рядки і стовпці на своєму місці, а також забезпечити відображення складних елементів управління всередині осередків. За допомогою Windows Forms можна легко створювати елементи управління з прив'язкою до даних. Створювати елементи управління з прив'язкою до даних можна шляхом перетягування об'єктів з допоміжного вікна в форми проекту.

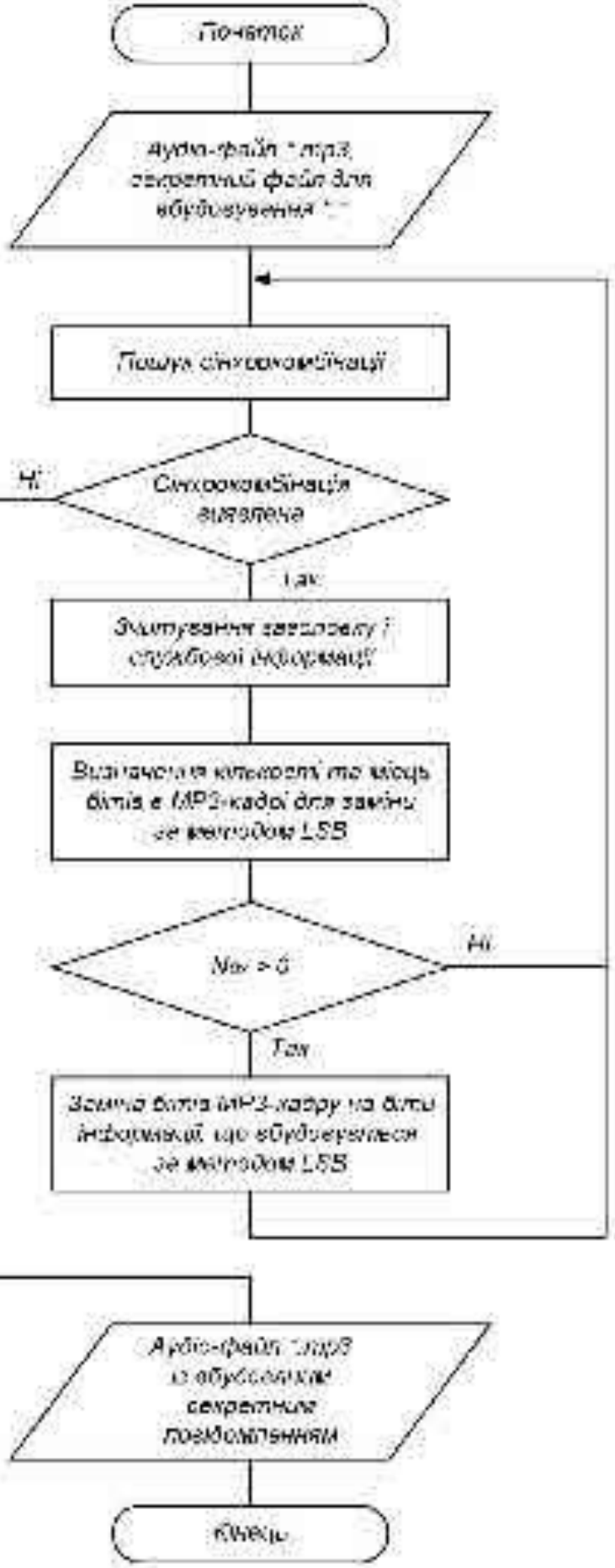


Рисунок 1.10. Блок-схема алгоритму приховування даних у аудіо-файлі

1.13 Реалізація програмного продукту

Схема структурна охоплює процес приховування даних в аудіо-файлі і наведена на рис. 1.10 у вигляді блок-схеми алгоритму.

В рамках даної роботи розроблено програмний додаток, що виконує приховування секретних даних у аудіо-файлі за допомогою модифікованого алгоритму LSB. На рисунку 1.11 показано початкову форму програмного продукту. На даному етапі користувачу потрібно обрати тип роботи програми – комплексне тестування якості приховування даних чи одноразове, тобто для одного аудіо-файлу.

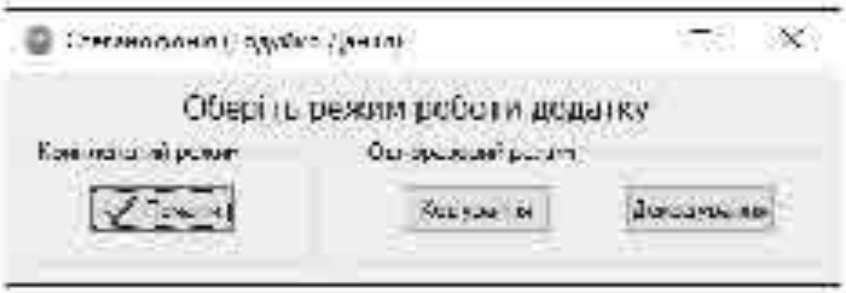


Рисунок 1.11. Початкова форма програмного продукту

На рисунку 1.12 показано наступну форму – після вибору варіанту приховування даних для одного файлу.

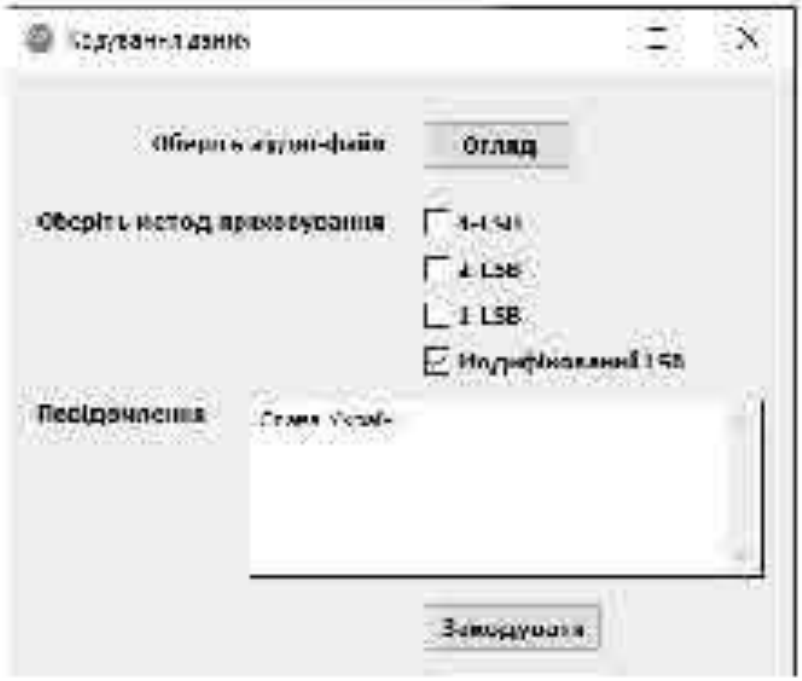


Рисунок 1.12. Форма кодування даних для одного файлу

На даному етапі потрібно обрати аудіо-файл, в який і буде приховано секретне повідомлення.

Рисунок 1.13. Форма з результатами приховування даних

Також, потрібно обрати метод, яким буде закодовано повідомлення до аудіо-контейнера та ввести секретне повідомлення. Програмний продукт дозволяє обрати одразу декілька методів, і тоді, в результаті кодування буде створено декілька стеганоконтейнерів.

На рисунку 1.13 показано результат кодування. На даному етапі користувач може прослухати стегано-контейнер, щоб спробувати відчути на власний слух чи з'явилися помітні зміни аудіо контейнеру. Також розраховуються коефіцієнти – відношення пікового сигналу відносно шумового співвідношення; подібність даних у відсотках між значенням контрольної суми стегано-контейнеру та вхідного файлу; подібність даних у відсотках між хеш-функцією стегано-контейнеру та вхідних даних, у відсотках; зміну частот у відсотках секретного повідомлення, витягнутого зі стегано-контейнера відносно вхідного секретного

повідомлення. Користувач може спотворити отриманий стегано-контейнер, додавши до нього адитивний гаусовий білий шум з певним коефіцієнтом дисперсії. Вихідні дані після кодування даних показані на рисунку 1.14.

Рисунок 1.14. Вихідні дані після кодування інформації

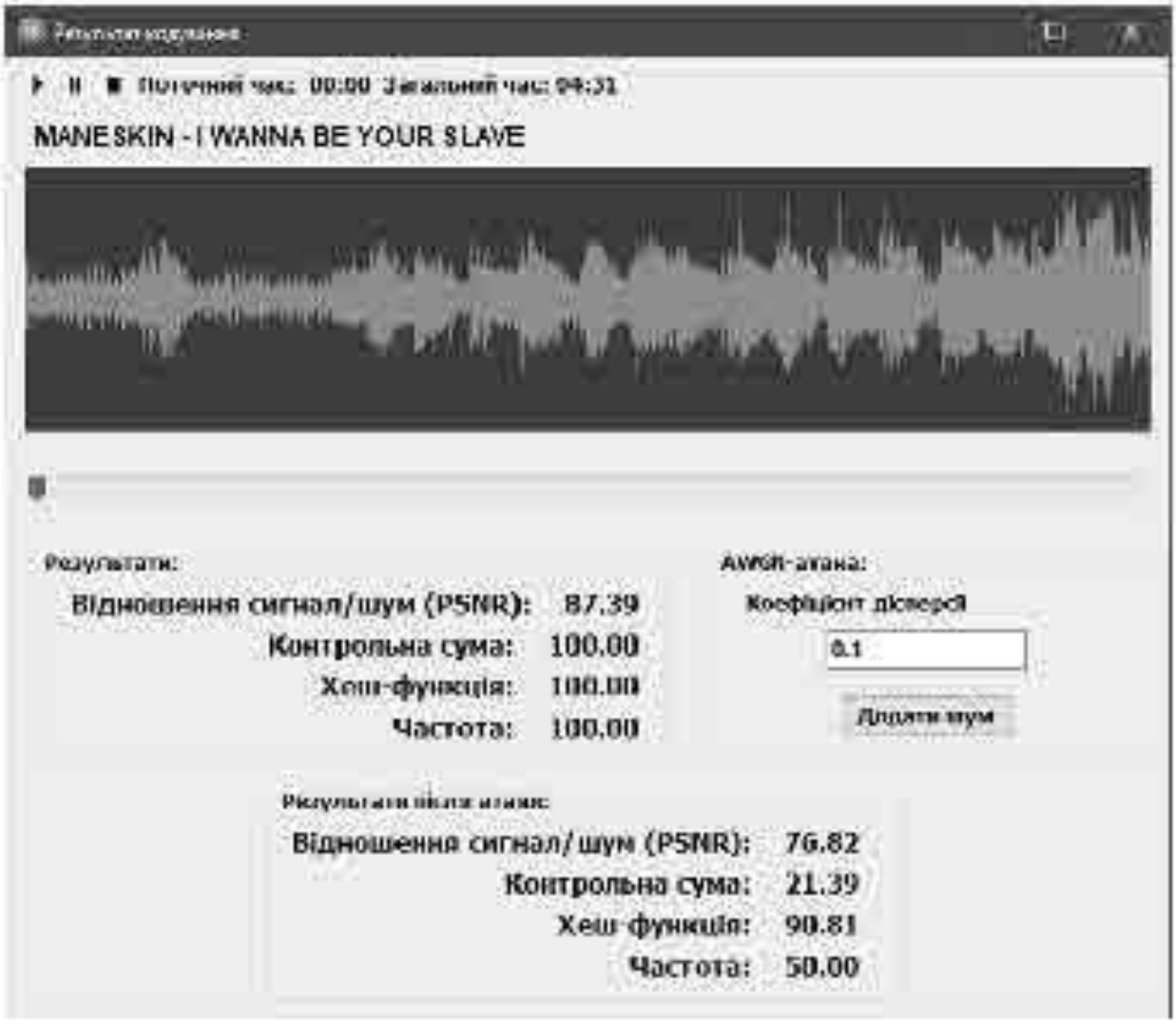


Рисунок 1.15. Видозмінена форма результату кодування після додавання атаки на стегано-контейнер

Після додавання атаки на стегано-контейнер отримані коефіцієнти перераховуються та додається ще один стегано-контейнер. Результат показано на рисунку 1.15. Додані вихідні дані після здійснення атаки на стегано-контейнер показаніна рисунку 1.16.

Рисунок 1.16. Вихідні даніпісля здійснення атаки на стегано-контейнер

Повертаючись до початкової форми, показаної на рисунку 1.11, оберемо декодування даних. Форма декодування даних показана на рисунку 1.17.

Рисунок 1.17. Форма декодування даних зі стегано-контейнеру

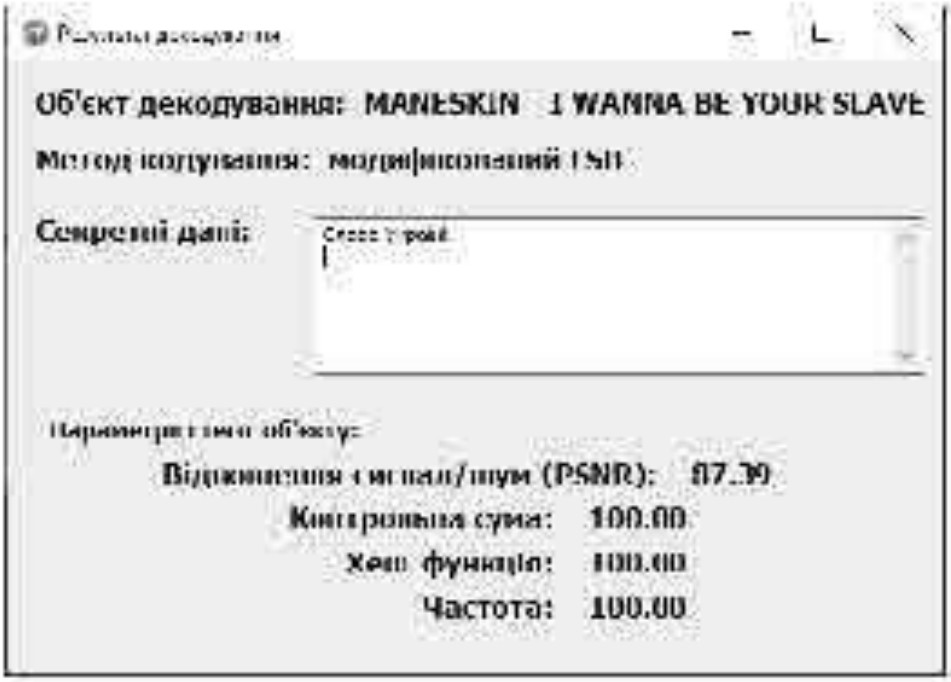


Рисунок 1.18. Результат декодування даних зі стегано-контейнеру

На даному етапі потрібно обрати стегано-контейнер із вкрапленим секретним повідомленням та обрати метод, яким було закодоване секретне повідомлення. Результат показано на рисунку 1.18. На формі, продемонстрованій вище, наведені назва стегано-контейнеру, з якого було отримане секретне повідомлення, саме секретне повідомлення, метод, яким секретне повідомлення було вкраплене у стегано-контейнер та параметри цього контейнеру. Обравши на початковій формі програмного продукту варіант комплексного дослідження (режим "Комплексний"), отримаємо форму, показану на рисунку 1.19:

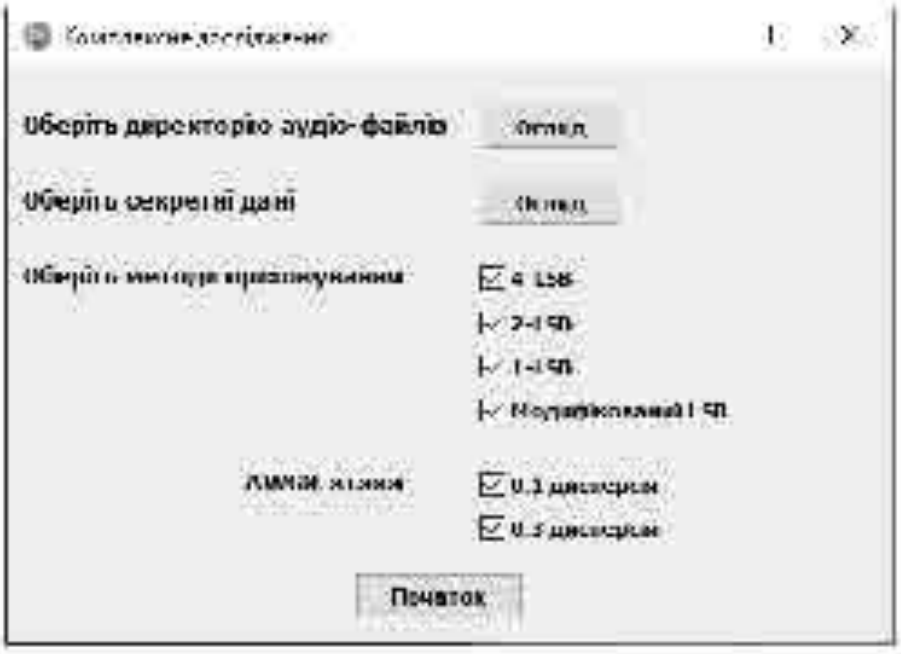


Рисунок 1.19. Форма налаштування комплексного дослідження

По-перше, потрібно обрати папку з вхідними даними – набір аудіофайлів, в які буде вкраплено приховані дані. Далі треба обрати секретні дані: секретними даними виступає також аудіо-файл, але з меншим бітрейтом – 96 кбіт/сек. Після цього треба обрати бажані методи вкраплення інформації та задати параметри атаки на отримані стегано-контейнери.

Всі результати дослідження формуються та записуються до Excel-файлу.

1.14 Аналіз результатів роботи програми

Проведена робота в даному дослідженні спрямована на ефективну модифікацію існуючого методу стеганофонії, а саме – методу найменш значущого біту (LSB). Крім того, пропонується ефективний спосіб перевірки

достовірності отриманих результатів за допомогою розробленого програмного продукту. Також, програмний застосунок надає зручні вихідні дані із всіма необхідними розрахунками та діаграмами. В проведеному дослідженні секретним повідомленням, яке вкраплювалось у вхідні дані, є аудіо файл формату MPEG Layer 3 (MP3). Для порівняння отриманих результатів, використовуючи розроблену модифікацію алгоритму найменш значущого біту, також проведені процеси приховування даних традиційними методами найменш значущого біту. Аудіо-повідомлення вбудовується у вхідні файли, використовуючи три традиційні методи LSB: 4-LSB, 2-LSB і 1-LSB та розроблену модифікацію, щоб порівняти їх ефективність.

1.14.1 Результати процесу приховування даних

У цій частині секретне повідомлення приховане в усіх вхідних файлах, використовуючи як модифікований алгоритм, так і традиційні методи LSB.

Таблиця 1.2. Результуючі значення PSNR для обраних методів

Стиль мелодії	*PSNR 4-LSB, Дб*	*PSNR 2-LSB, Дб*	*PSNR 1-LSB, Дб*	*PSNR модифікації LSB, Дб*
Класична	35,15	45,78	57,52	75,21
Джаз	47,26	70,12	83,48	102,12
Кантрі	42,47	60,15	74,31	92,84
R&B	46,94	68,85	82,54	101,98
Реп	38,41	58,54	64,87	79,54
Реггі	40,51	62,48	71,12	90,26
Поп	29,05	51,96	57,96	73,47
Рок	41,98	66,21	80,25	99,09
Блюз	44,14	63,88	76,12	95,87
Хіп-хоп	33,48	54,63	62,15	77,32

В цьому дослідженні вхідними даними виступають по 20 MP3-файлів кожного із наступних стилів музики: Класична, Джаз, Кантрі, R&B, Реп, Реггі, Поп, Рок, Блюз, Хіп-хоп. Результуючі значення PSNR оцінені після вставки секретного повідомлення, розміром близько 1 Мбайт для 10 різних стилів музики. Секретне повідомлення приховане за допомогою розробленої модифікації та трьох традиційних методів LSB – 4-LSB, 2- LSB і 1-LSB. Результати показані в таблиці 1.2 та на рисунку 1.20.

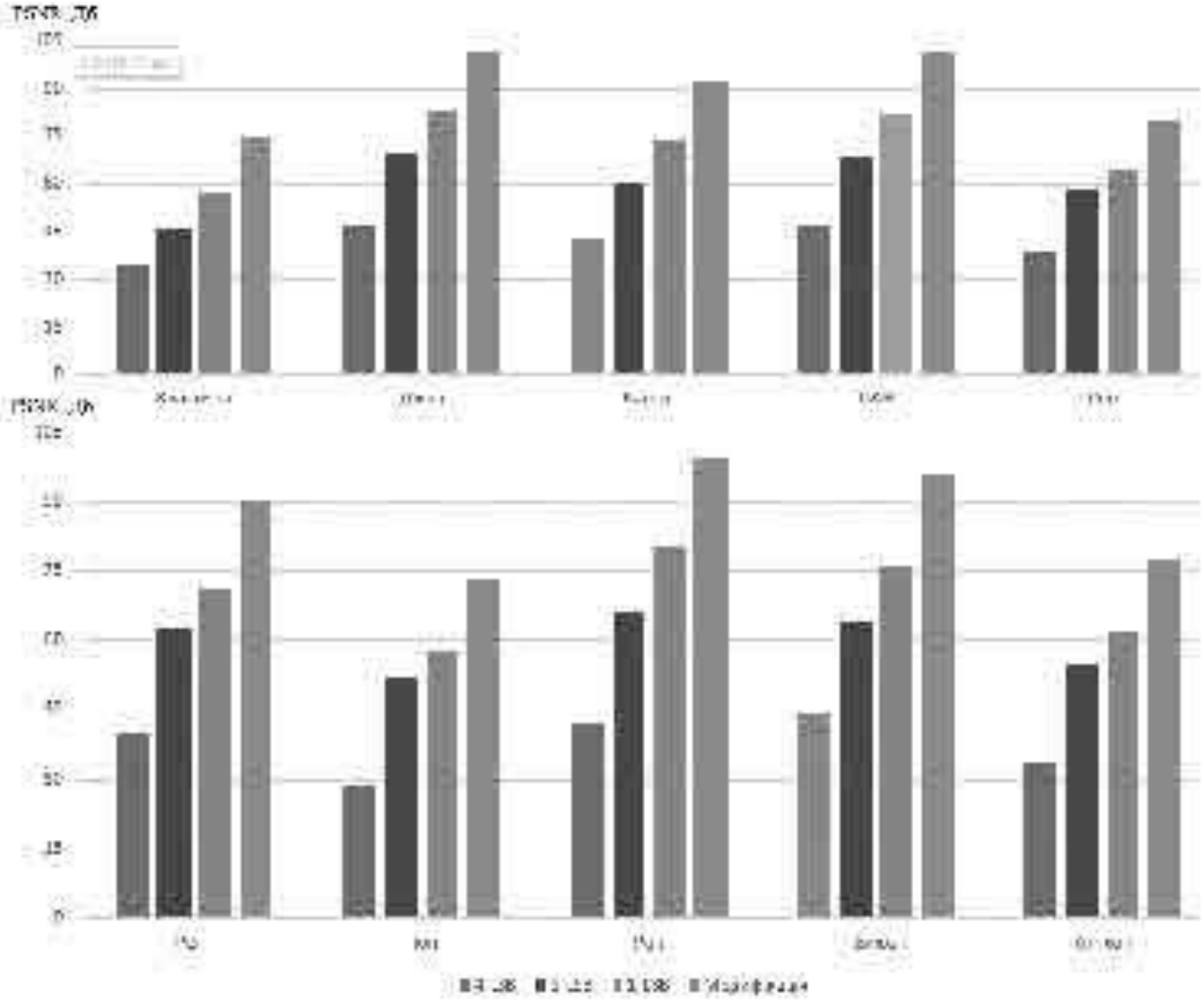


Рисунок 1.20. Результуючі значення PSNR для обраних методів

Можна чітко бачити, що результати PSNR пропонованого методу краще, ніж традиційні LSB для всіх жанрів. Як показано вище, Джаз показує найнижчий PSNR, оскільки він є одним з MP3-файлів, який містить більш високий рівень шуму, ніж інші звукові файли.

У наведеній нижче таблиці 13 показаний відсоток покращення між поточним методом та традиційними методами для попередніх вхідних даних на основі отриманих даних на попередньому кроці.

1.14.2 Результати додавання AWGN-атаки

У цій частині дослідження застосовується один з видів атаки – адитивний гаусовий білий шум (AWGN) додається до стегано-контенерів, тобто для всіх вихідних даних, які були отримані модифікованим алгоритмом, перш ніж

витягати з них секретне повідомлення. Потім повідомлення витягується та здійснюється порівняння його оцінки PSNR зі значенням цієї оцінки стегано-контейнерів до застосування атаки.

Таблиця 1.3. Перевага модифікованого алгоритму відносно традиційних

Стиль мелодії	*Відношення PSNR 4-LSB до модифікації, %*	*Відношення PSNR 2-LSB до модифікації, %*	*Відношення PSNR 1-LSB до модифікації, %*
Класична	53,26	39,13	23,52
Джаз	53,72	31,34	18,25
Кантрі	54,25	35,21	19,96
R&B	53,97	32,49	19,06
Реп	51,71	26,40	18,44
Реггі	55,12	30,78	21,21
Поп	60,46	29,28	21,11
Рок	57,63	33,18	19,01
Блюз	53,96	33,37	20,60
Хіп-хоп	56,70	29,35	19,62

Атака AWGN додана до всіх стегано-контейнерів, які отримали секретні дані розробленим модифікованим алгоритмом з різними значеннями шумової дисперсії. Таблиця 1.4 показує отримані значення PSNR для стегано-контейнерів після атаки зі значенням дисперсії 0,1 біт/сек/Гц для всієї смуги кожного стегано-контейнеру.

Таблиця 1.4. Результати після атаки до стегано-контейнерів зі значенням дисперсії 0.1

Стиль мелодії	*PSNR до атаки, Дб*	*Дисперсія, біт/сек/Гц*	*PSNR після атаки, Дб*	*Погіршення, %*
Класична	75,21	0,1	69,76	7,25
Джаз	102,12	0,1	95,33	6,65
Кантрі	92,84	0,1	86,83	6,47
R&B	101,98	0,1	97,92	3,98
Реп	79,54	0,1	75,32	5,31
Реггі	90,26	0,1	85,56	5,21
Поп	73,47	0,1	70,52	4,02
Рок	99,09	0,1	93,82	5,32
Блюз	95,87	0,1	92,56	3,45
Хіп-хоп	77,32	0,1	73,57	4,85

У таблиці 1.5 показані досягнуті значення PSNR для стегано-контейнерів після атаки зі значенням дисперсії 0,3 біт/сек/Гц для всієї смуги кожного стегано-контейнеру.

Таблиця 1.5. Результати після атаки до стегано-контейнерів зі значенням дисперсії 0.3

Стиль мелодії	*PSNR до атаки, ДБ*	*Дисперсія, біт/сек/Гц*	*PSNR після атаки, ДБ*	*Погіршення, %*
Класична	75,21	0,3	65,92	12,35
Джаз	102,12	0,3	92,58	9,34
Кантрі	92,84	0,3	83,33	10,24
R&B	101,98	0,3	94,70	7,14
Реп	79,54	0,3	72,41	8,96
Реггі	90,26	0,3	81,61	9,58
Поп	73,47	0,3	67,88	7,61
Рок	99,09	0,3	89,60	9,58
Блюз	95,87	0,3	89,19	6,97
Хіп-хоп	77,32	0,3	71,05	8,11

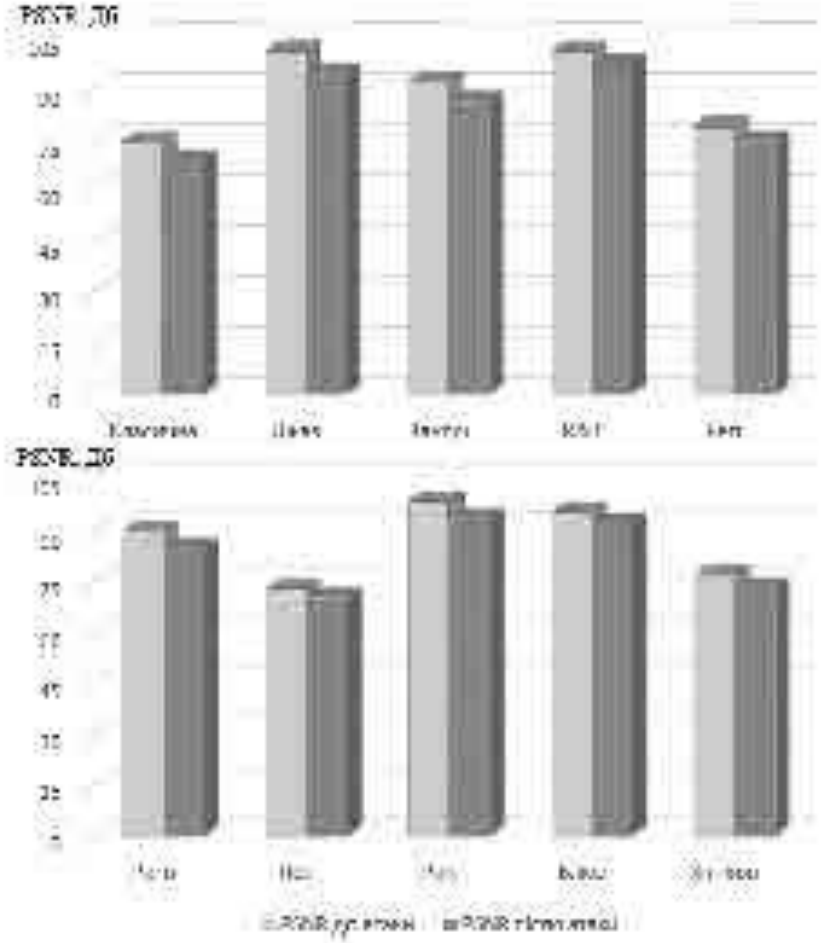


Рисунок 1.21. Порівняння значень PSNR до і після додавання атаки зі значенням шумової дисперсії 0,1 біт/сек/Гц

Можна зробити висновок про наявність очевидної деградації значення PSNR після додавання атаки AWGN. Як показано в таблицях вище, деградація у значеннях PSNR збільшується із збільшенням значення дисперсії шумів. Отримані результати значень PSNR після додавання атаки AWGN з шумовою дисперсією 0,1 показані на рисунку 1.21.

Отримані результати значень PSNR після додавання атаки AWGN з шумовою дисперсією 0,3 показані на рисунку 1.22.

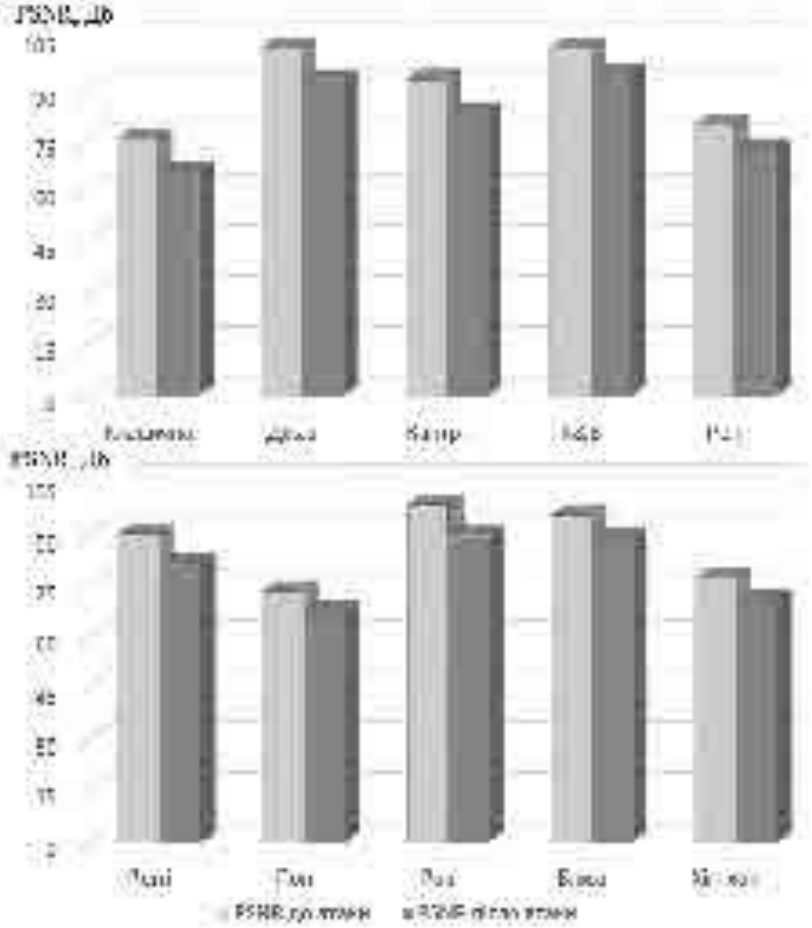


Рисунок 1.22. Порівняння значень PSNR до і після додавання атаки зі значенням шумової дисперсії 0,3 біт/(вибірка/Гц)

1.14.3 Результати розрахунку показників цілісності

Для порівняння стегано-контейнерів з вхідними файлами розроблений додаток розраховує наступні показники: значення контрольної суми стегано-контейнеру та вхідного файлу з видачею значення подібності даних у відсотках; розраховує хеш-функції стегано-контейнерів та вхідних даних, а також показує у відсотках, наскільки два файли подібні між собою; зміна частот показує у

відсотках зміну секретного повідомлення, витягнутого зі стегано-контейнера відносно вхідного секретного повідомлення. У наведеній нижче таблиці 1.6 представлені досягнуті відсотки для вихідних даних, отриманих модифікованим алгоритмом відносно вхідних даних, до застосування атаки на них.

Таблиця 1.6. Результати показників цілісності для модифікації алгоритму

Стиль мелодії	*Контрольна сума, %*	*Хеш-функція, %*	*Зміна частот, %*
Класична	100	100	100
Джаз	100	100	100
Кантрі	100	100	100
R&B	100	100	100
Реп	100	100	100
Реггі	100	100	100
Поп	100	100	100
Рок	100	100	100
Блюз	100	100	100
Хіп-хоп	100	100	100

Як показано у таблиці вище, розроблена модифікації алгоритму не спотворює вихідні дані, оскільки всі показники оцінки якості отриманих даних мають значення 100% для всіх стилів музики. Ці результати отримані без застосування атаки на стегано-контейнери. Далі виконується процес атаки з додаванням гаусового білого шуму (AWGN) до стегано-контейнерів в результаті розробленої модифікації алгоритму. Результати, отримані у процесі порівняння стегано-контейнерів до атаки та після за цими критеріями, показаніу таблиці 1.7.

Таблиця 1.7. Результати показників цілісності після застосування атаки

Стиль мелодії	*Контрольна сума, %*	*Хеш-функція, %*	*Зміна частот, %*
Класична	21,42	92,14	50
Джаз	21,25	91,88	50
Кантрі	21,71	92,54	50
R&B	21,14	92,17	50
Реп	20,84	91,25	50
Реггі	20,54	90,28	50
Поп	20,65	90,77	50
Рок	21,21	91,25	50
Блюз	20,25	90,17	50
Хіп-хоп	21,17	90,14	50

Як показано в таблиці вище, найкращий відсоток подібності стегано-контейнерів до атаки та після, демонструє перевірка хеш-функції, тоді як мінімальні досягнуті результати є для значення контрольної суми. Графічне відображення отриманих результатів, з таблиці 1.7 продемонстровано на рисунку 1.23.

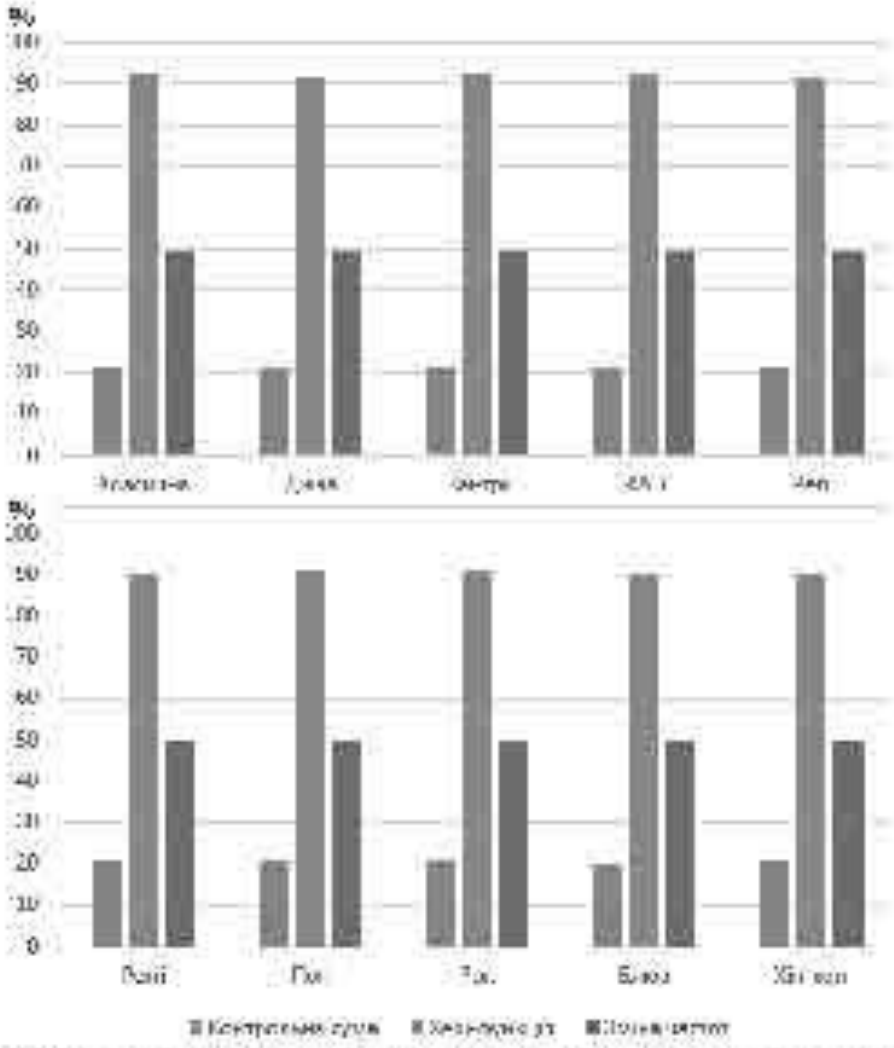


Рисунок 1.23. Показники цілісності стегано-контейнерів після застосування атаки

Окрім наведених вище результатів розрахунку показників цілісності проведено оцінювання показників цілісності для традиційного 1-LSB методу. У цьому випадку атака застосовується до стегано-контейнерів отриманих методом 1-LSB. У таблиці 1.8 показані досягнуті значення PSNR для стегано-контейнерів після атаки зі значенням дисперсії 0,1 біт/кекГц для всієї смути кожного стегано-контейнеру. Таблиця показує, що після атаки, тобто додавання гаусового білого шуму (AWGN) до стегано-контейнерів, отриманих методом 1-LSB, спостерігається деградація значень PSNR. Крім того, очевидно, що деградація

PSNR набагато більша порівняно з розробленою модифікацією алгоритму.

Таблиця 1.8. Результати після атаки до стегано-контейнерів зі значенням дисперсії 0.1

Стиль мелодії	*PSNR до атаки, Дб*	*Дисперсія, біт/сек/Гц*	*PSNR після атаки, Дб*	*Погіршення, %*
Класична	57,52	0,1	48,34	15,96
Джаз	83,48	0,1	71,15	14,77
Кантрі	74,31	0,1	66,14	10,99
R&B	82,54	0,1	72,58	12,07
Реп	64,87	0,1	53,74	17,16
Реггі	71,12	0,1	60,59	14,81
Поп	57,96	0,1	49,15	15,20
Рок	80,25	0,1	69,54	13,35
Блюз	76,12	0,1	64,97	14,65
Хіп-хоп	62,15	0,1	50,85	18,18

У таблиці 1.9 показані результати, отримані у процесі порівняння стегано-контейнерів до атаки та після для техніки 1-LSB. Як показано в таблиці, технологія 1-LSB пропонує гірші значення показників цілісності стегано-контейнерів після застосування атаки на них, в порівнянні з розробленою модифікацією алгоритму.

Таблиця 1.9. Результати показників цілісності після застосування атаки для методу 1-LSB

Стиль мелодії	*Контрольна сума, %*	*Хеш-функція, %*	*Зміна частот, %*
Класична	17,58	82,14	44,47
Джаз	19,58	86,88	44,47
Кантрі	18,25	80,54	44,47
R&B	20,21	79,17	44,47
Реп	19,28	85,25	44,47
Реггі	18,17	88,28	44,47
Поп	18,25	84,77	44,47
Рок	17,85	89,25	44,47
Блюз	20,85	83,17	44,47
Хіп-хоп	19,57	84,14	44,47

Під час тестування роботи програмного забезпечення збоїв та недоліків не виявлено, що говорить про можливість впровадження програмного продукту за потреби.

2 ЕКОНОМІЧНА ЧАСТИНА

Метою даних розрахунків є обчислення вартості виконання науково-дослідної роботи ««Розробка алгоритмічного та програмного забезпечення для стеганофонії». Оцінка якості розробленого проекту включає визначення трудомісткості і вартості його створення. У технологічній структурі науково-дослідних робіт виділяємо декілька самостійних етапів, а саме: розробка технічного завдання, вибір напряму дослідження, теоретичні і експериментальні дослідження, узагальнення і оцінка результатів.

Розрахунок трудомісткості НДР здійснений в наступній послідовності:

1) Складений перелік всіх етапів і видів робіт, які необхідно виконати в ході даної НДР. Після узгодження з керівником проекту допущено виключення, доповнення, об'єднання окремих етапів і видів робіт;

2) По кожному виду робіт визначений кваліфікаційний рівень виконавців. В разі виконання однієї роботи виконавцями різної кваліфікації, роботу розподілили на ряд паралельних конкретних робіт для кожної категорії виконавця. Перелік етапів і робіт, що виконуються при проведенні НДР, приведений в таблиці 2.1.

Таблиця 2.1 Розподіл робіт по етапах і видах виконавців

Етап проведення НДР	Вигляд робіт	Посада виконавця
Розробка технічного завдання (ТЗ)	1.Складання і затвердження ТЗ для НДР «Розробка алгоритмічного та програмного забезпечення для стенографії»	Дипломник, керівник
Вибір напряму дослідження	1. Збір і вивчення науково-технічної літератури, технічної документації і інших матеріалів. 2. Формулювання можливих напрямів вирішення завдань, поставлених в технічному завданні НДР і їх порівняльна оцінка.	Дипломник керівник

	3. Вибір напряму проведення досліджень для подальшої розробки. 4. Розробка плану проведення досліджень для подальшої розробки.	
Теоретичні і експериментальні дослідження	1.Аналітичний огляд методів стеноргафії 2. Модифікація алгоритму LSB 3. Дослідження методів аудіостеноргафії 4. Практична реалізація програмного продукту та аналіз результатів	Дипломник керівник консультанти
Узагальнення і оцінка результатів досліджень	1. Узагальнення результатів 2. Оцінка повноти вирішення поставлених завдань. 3. Складання і оформлення звіту. Розгляд результатів проведеною НДР і прийняття результатів в цілому.	Дипломник керівник консультанти

В умовах відсутності нормативної бази тривалість виконання окремих робіт розраховуємо на основі вірогідних оцінок робіт, що задаються виконавцями.

Таблиця 2.2 Очікувана трудомісткість робіт

Вид діяльності роботи	Очікуваний час виконання (дні)
1. Складання і затвердження ТЗ для НДР «Розробка алгоритмічного та програмного забезпечення для стеноргафії»	1
2. Збір і вивчення науково – технічної літератури, технічної документації і інших матеріалів.	2
3. Формулювання можливих напрямів вирішення завдань, поставлених в технічному завданні НДР і їх порівняльна оцінка.	1
4. Вибір напряму проведення досліджень і способів вирішення поставлених завдань. Розробка плану проведення досліджень	1

для подальшої розробки.	
5. Аналітичний огляд методів стеганографії	5
6. Модифікація алгоритму LSB	6
7. Дослідження методів аудіостеганографії	6
8. Практична реалізація програмного продукту та аналіз результатів	
Всього:	22

Результатом виконання НДР є науково-технічна продукція, що є закінчені науково – дослідницькі роботи, виконані відповідно до вимог, передбачених договором, і прийнятими замовником. Виходячи з особливостей створення науково – технічної продукції і її залежності від інтелектуальної праці, розрахунок собівартості і ціни виконання НДР включає наступні статті витрат: витрати на матеріали, основна і додаткова заробітна плата, відрахування до єдиного соціального фонду страхування, витрати на роботи, що виконуються сторонніми організаціями, і деякі інші.

1) Витрати на матеріали складаються з вартості придбаного паперу формата А4 і становитимуть 198 грн.

2) До витрат «Основна заробітна плата» відносяться оплата праці виконавців, безпосередньо причетних до її виконання. Розмір основної зарплати встановлюється виходячи з чисельності різних категорій виконавців, трудомісткості, що витрачається ними на виконання різних видів робіт, а також їх середньої заробітної плати (ставки) за один робочий день. Відповідно до статті 8 «Закону про Державний бюджет України на 2022» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2022 року - 6500 гривень; мінімальну погодинну тарифну ставку – 39,26 грн.

Середня зарплата за один робочий день для кожного виконавця визначена по формулі:

$$З_{ден} = п т с. * 8;$$

де п т с – погодинна тарифна ставка, грн..;

8 – тривалість робочого дня, год.

Зден дипломника = 39.26*8 = 314,08 грн.

Зден керівника = 55.00 *8 = 440 грн.

Зден консультантів = 55.00 *8 = 440 грн.

Витрати на основну заробітну плату, НДР, що включаються в собівартість, приведені в таблиці 2.3.

Таблиця 2.3 Витрати на основну заробітну плату

Виконавець	Погодинна тарифна ставка, грн	Денна ставка, грн	Трудомісткість робочих днів	Сума основної зарплати, грн
Дипломник	39,26	314,08	22	6512
Керівник	55,00	440	1	440
Консультант по економічній частині	55,00	440	0,25	110
Консультант по охороні праці	55,00	440	0,25	110
Нормоконтроль	55,00	440	0,25	110
Всього (Зо)				7282

3) Витрати на додаткову заробітну плату визначаються у відсотках від основної. У наукових закладах додаткова заробітна плата складає 10-12% від основної заробітної плати.

$$Зд = 10\% Зо = 7282 * 0,1 = 728,2 \text{ грн}$$

4)До складу собівартості НДР включаються податки, збори і інші обов'язкові платежі, встановлені системою оподаткування що діє. Відрахування до єдиного соціального внеску складає:

$$Зесв = 0,22 * (Зо + Зд) = 0,22 * (7282 + 728,2) = 1762,24 \text{ грн}$$

5) До накладних витрат відносять витрати на управління і господарське обслуговування, що відносяться до всіх виконуваних НДР. У наукових закладах накладні витрати складають 40 - 120% від основної і додаткової заробітної плати.

$$Рнакл = (Зо + Зд) * 0,5 = (7282 + 728,2) * 0,5 = 4005,10 \text{ грн.}$$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому НДР за формою, приведеною в таблиці 2.4.

Таблиця 2.4 Калькуляція планової собівартості

Статті витрат	Сума, грн.
1. Матеріали	198,00
2. Основна заробітна плата	7282
3. Додаткова заробітна плата	728,20
4. Відрахування до єдиного соціального внеску	1762,24
5. Накладні витрати	4005,10
Планова собівартість (Спл)	13975,54

Плановий прибуток визначений по формулі:

Ппл = 0,1 *Спл = 0,1*13975,54 = 1397,55 грн.

Де 0,1 – норматив, який враховує граничний рівень рентабельності, встановлений чинним законодавством для науково-технічної продукції.

Договірна ціна визначається по формулі:

Цдір = Спл + Ппл = 13975,54 + 1397,55 = 15373,09 грн.

Ціну реалізації встановлюємо з урахуванням ПДВ

Цр = Цдір + ПДВ; Цр = 15373,09 +0,2*15373,09 = 18447,71грн.

3 ОХОРОНА ПРАЦІ

Безпека праці, спрямована на створення небезпечних і нешкідливих умов праці. На сучасному етапі розвитку виробництва вона набуває все більше важливого значення.

Вирішення завдань охорони праці базується на досягненнях ергономіки, наукової організації праці, технічної естетики, гігієни та фізіології праці, психофізіології. Крім того, успіх охорони праці визначається темпами впровадження передової техніки, підвищення рівня механізації і автоматизації виробничих процесів, удосконаленням технології та організації виробництв

Безпека праці на підприємстві може бути на належному рівні тільки тоді, коли всебічно відповідає вимогам трудового законодавства, державним стандартам України, норм і правил, розроблених для збереження здоров'я працюючих. Важливе місце при цьому належить виконанню організаційних вимог з охорони праці, а також трудовій та виробничій дисципліні працюючих.

Дипломний проектом передбачена розробка алгоритмічного та програмного забезпечення для стегафонії. Виконання даної роботи проводилося за допомогою персонального комп'ютера. У зв'язку з цим необхідно проаналізувати фактори ризику при роботі з сучасним персональним комп'ютером.

3.1 Аналіз небезпечних та шкідливих чинників, що впливають на працівника

Основними факторами шкідливого впливу ПК на організм людини є такі:

1. Електромагнітні поля;
2. Електромагнітні випромінювання;
3. Розгортка зображення на моніторі;
4. Мелькання зображення на екрані;
5. Тривала нерухомість пози оператора.

Сукупний вплив на людину всіх шкідливих факторів знижує загальний біоенергетичний потенціал і опірність організму, знижує імунітет, викликає

м'язову атрофію і застої в органах. Наслідки порушення норм безпеки при роботі за ПК можуть викликати професійні захворювання або призвести до нещасного випадку та травмування працівника.

3.2 Розробка заходів з охорони праці

Зменшити вплив перерахованих факторів ризику і зберегти здоров'я людині, яка постійно використовує в роботі ПК, дозволяє дотримання всіх заходів і засобів, передбачених охороною праці.

3.2.1 Мікроклімат робочої зони працівників, вентиляція

У виробничих приміщеннях на робочих місцях з ВДТ мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості і рухливості повітря (ГОСТ 12.1.005-88, СН 4088-86).

Рівні позитивних і негативних іонів в повітрі приміщень з ВДТ повинні задовольняти санітарно-гігієнічним нормам №2152-80

3.2.2 Освітлення робочого місця, шум, вібрація

Штучне освітлення в приміщеннях з робочими місцями, обладнаними ЕОМ і ПЕОМ, має здійснюватись системою загального рівномірного освітлення. Значення освітленості на поверхні робочого столу в зоні розміщення документів має становити 300 - 500 лк.

Як джерело світла при штучному освітленні застосовуються переважно люмінесцентні лампи.

Значення освітленості на поверхні робочого столу в зоні розміщення документів має становити 300 - 500 лк.

Система загального освітлення має становити суцільні або переривчасті лінії світильників, розташовані збоку від робочих місць (переважно зліва), паралельно лінії зору працюючих. Застосування світильників без розсіювачів та екрануючих ґрат заборонено.

Рівні звукового тиску в октавних смугах частот мають відповідати вимогам СН 3223-85, ГОСТ 12.1.003-83, ГР 2411-81.

Для забезпечення допустимих рівнів шуму на робочих місцях слід застосовувати засоби звукопоглинання, При виконанні робіт з ЕОМ у виробничих приміщеннях значення характеристик вібрації на робочих місцях не повинні перевищувати допустимі згідно СН 3044-84, ГОСТ 12.1.012-90. При розумовій праці, яка вимагає зосередженості припустимий рівень шуму становить 50дБ

3.2.3 Організація робочого місця користувача ПК

- Важливо, щоб офісний працівник сидячи за комп'ютером знаходився за добре освітленим робочим столом. Найчастіше саме погане освітлення робочого місця надає більш згубний для зору вплив, ніж сам факт перебування за комп'ютером.
- Робочі столи слід розміщувати таким чином, щоб монітори були орієнтовані бічною стороною до світлових прорізів, щоб природне світло падало переважно ліворуч.
- При розміщенні робочих місць відстань між робочими столами повинна бути не менше 2,0 м, а відстань між бічними поверхнями відеомоніторів - не менше 1,2 м.
- Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання.
- Конструкція робочого стільця або крісла повинна забезпечувати підтримку раціональної робочої пози працівника.
- Клавіатуру слід розташовувати на поверхні столу на відстані 100.300 мм від краю, зверненого до користувача, або на спеціальній поверхні, відокремленій від основної стільниці.
- Екран відеомонітора повинен знаходитися від очей користувача на відстані 600-700 мм, але не ближче 500мм.

Безпека праці при роботі за комп'ютером передбачає, що тривалість безперервної роботи за комп'ютером без регламентованої перерви не повинна перевищувати 2 години.

Не рекомендується працювати за комп'ютером більше 6 годин за зміну. Рекомендується робити перерви в роботі за ПК тривалістю 10 хвилин через кожні 50 хвилин роботи. Під час регламентованих перерв доцільно виконувати комплекси вправ.

При нерегламентованій роботі підвищеної інтенсивності можливі головні болі, нервові зриви та інше.

3.3 Пожежна безпека

Протипожежна безпека на підприємстві – невіддільна частина організації робочого простору і процесів згідно з нормами чинного законодавства. Зокрема, цю сферу регламентують Правила пожежної безпеки в Україні, затверджені наказом Міністерства внутрішніх справ України, зі змінами, які періодично вносяться відповідними наказами.

Попри обладнання будівель будь-якими типами установок пожежогасіння, пожежної сигналізації або внутрішніми пожежними кранами, офісні приміщення також мають бути забезпечені первинними засобами пожежогасіння.

До первинних засобів пожежогасіння належать: вогнегасники, кошма (покривало з негорючого теплоізоляційного полотна), ящики з піском, бочки з

водою, пожежні відра, багри, ломи, совки тощо. Найбільш зручними для використання є вогнегасники.

Відповідальними за своєчасне та повне оснащення об'єктів засобами пожежогасіння, забезпечення їх технічного обслуговування, навчання працівників правил користування ними є роботодавець та керівники структурних підрозділів.

Відповідальні особи зобов'язуються розробити протипожежний режим і інструкції відповідно до вимог, викладених в нормативних актах на зазначених їм об'єктах.

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх користування та утримання, наприклад:

- евакуаційних шляхів,
- так званих «курилок»,
- місць складування продукції та сировини,
- стоянки транспорту.

Важливою складовою протипожежного режиму на будь-якому об'єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам.

ВИСНОВКИ

У даному проекті розроблено алгоритмічне та програмне забезпечення для стеганофонії. При цьому виконано модифікацію алгоритму найменш значущого біту (LSB). Дана модифікація показала кращі результати ніж традиційні методи LSB, а також продемонструвала вищий рівень стійкості до атаки з додаванням адитивного гаусового білого шуму.

Під час виконання роботи було проаналізовано предметну галузь та розглянуто існуючі методи стеганофонії, проаналізовано їх переваги та недоліки, проведене їх порівняння. Було проведено аналіз зменшення внесення помітних змін до файлів-контейнерів, та щодо збільшення стійкості контейнерів до атак.

На основі проведеного аналізу було вирішено розробляти модифікацію саме методу найменш значущого біту через те, що цей метод забезпечує більшу безпеку та є ефективним способом приховування секретної інформації від хакерів і відправлення у пункт призначення безпечним та невиявленим способом. Також метод гарантує, що розмір файлу не змінюється навіть після кодування і також підходить для будь-якого типу формату аудіо-файлів. Він дозволяє приховувати у файлах-контейнерах набагато більший об'єм секретної інформації у порівнянні з іншими алгоритмами.

Для перевірки ефективності розробленого алгоритму було проведено дослідження на аудіо-файлах різних жанрів, рівного розміру. Також проведено імітацію атаки на ці контейнери для перевірки їх стійкості у порівнянні з традиційними методами найменш значущого біту. При порівнянні з традиційним алгоритмом LSB показано, що розроблена модифікація має кращі результати ефективності.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Конахович Г.Ф., Пузиренко А.Ю. Комп'ютерна стеганографія. Теорія та практика. [Текст] / Конахович Г.Ф // Київ: МК-Пресс., 2006 р. – 288 с.
2. Грибунин В.Г., Оков И.Н., Туринцев И.В Цифровая стеганография [Текст] / Грибунин В.Г. // Москва: СОЛОН-Пресс, 2002 р. – 261 с.
3. Поліновський В.В. Інформаційна технологія для досліджень методів стеганографії і стеганоаналізу / В.В. Поліновський, В.Ю. Корольов, В.А. Герасименко, М.Л. Горинштейн // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2011. – №5. – С. 236–242.
4. Кошкина Н.В. Внедрение ЦВЗ в аудиосигналына основе пакетной вейвлет-декомпозиции и частотного маскирования / Н.В. Кошкина // Искусственный интеллект. – 2010. – № 4. – С.381–387.
5. Кошкина Н.В. Исследование применимости матрицы смежности для выявления стеганоаудиоконтейнеров / Н.В. Кошкина // Междунар. научно-техн. журнал «Проблемы управления и информатики». – 2014. – №1 – С. 148–156.
6. Обзор методов решений аудио стеганографии [Електронний ресурс] / Режим доступу: https://sibac.info/studconf/tech/xlii/54331.
7. Кошкіна Н.В. Новий метод цифрових водяних знаків для аудіосигналів / Н.В. Кошкіна // Матеріали 1 Міжнар. наук.-техн. конф. «Захист інформації і безпека інформаційних систем». – Львів. – 2012. – 120–121.
8. Кудин А.М. Математическая модель стеганографической системы на базе общей теории оптимальных алгоритмов/ А.М. Кудин // Математичне та комп'ютерне моделювання: Технічні науки. – 2010. – Вип. 4. – С. 136 – 143.
9. Задирака В.К. К вопросу стойкости стеганосистемы при пассивных атаках / В.К. Задирака, Л.Л. Никитенко // Междунар. научно-техн. журнал «Проблемы управления и информатики». – 2009. – № 2. – С. 138 – 1.

ДОДАТОК А. Лістинг основних класів ПЗ стеганофонії

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Runtime.Serialization;
using System.Text;
using Steganography.Media;

namespace Steganography
{
  [Serializable]

  //у цьому класі приховуємо та витягаємо повідомлення, що шифрується
  class HideAndExtract
  {
    private WavAudio _file;
    private List<byte> _bits;

    //ініціалізуємо об'єкт класу за допомогою класу WavAudio
    public HideAndExtract(WavAudio file)
    {
      _file = file;
    }

    //приховуємо повідомлення у лівому та правому потоках аудіо-файлу
    public void HideMessage(string message)
    {
      //отримуємо канали з файлу WaveAudio
      List<short> leftStream = _file.GetLeftStream();
      List<short> rightStream = _file.GetRightStream();

      //приховуємо повідомлення у потоках

      //перетворюємо повідомлення у масив байтів
      byte[] bufferMessage = Encoding.UTF8.GetBytes(message);
      short tempBit;
      //місце - індекс, який буде йти за повідомленням
      int bufferIndex = 0;
      //довжина повідомлення
      int bufferLength = bufferMessage.Length;
          //довжина аудіо (довжина лівого каналу дорівнює довжині правого каналу)
      int channelLength = leftStream.Count;
          //блок збереження повідомлення. Це значення дорівнює 1. Якщо воно
          //не дорівнює 1, то довжина повідомлення, що шифрується, більше,
          //ніж довжина початкового аудіофайлу
          int storageBlock = (int)Math.Ceiling((double)bufferLength / (channelLength * 2));

      //якщо довжина повідомлення більше, ніж довжина аудіоканалу
```

```
    if (bufferLength > channelLength)
      throw new Exception();

          //зберігаємо інформацію про довжину повідомлення, що шифрується,
          //у перших елементах лівого і правого потоків аудіо-файлу

          //беремо цілу частину розміру повідомлення і записуємо першим
          //елементом у лівому каналі
    leftStream[0] = (short)(bufferLength / 32767);
          //беремо залишок розміру повідомлення і записуємо першим
          //елементом у правому каналі
    rightStream[0] = (short)(bufferLength % 32767);
    var countBufferMessage = 0;
          //йдемо по довжині потоку, починаючи з 1, тому що у [0]
          //зберігається довжина повідомлення; зберігаємо біт повідомлення
          //у лівий і правий потоки
    for (int i = 1; i < leftStream.Count && countBufferMessage < bufferMessage.Length; i++)
    {
          //беремо залишок від ділення на 8, тому що працюємо з бітами,
          //так як працюємо з цифрою 7, тому що діапазон складає [0..7]
      if (bufferIndex < bufferLength && i % 8 > 7 - storageBlock && i % 8 <= 7)
      {
        //отримуємо біт повідомлення
        tempBit = bufferMessage[bufferIndex++];
        //заміщуємо біт аудіоданих бітом повідомлення
        leftStream.Insert(i, tempBit);
        leftStream[i] = tempBit;
        countBufferMessage++;
      }

      if (bufferIndex < bufferLength &&  i % 8 > 7 - storageBlock && i % 8 <= 7)
      {
        tempBit = bufferMessage[bufferIndex++];
        rightStream.Insert(i, tempBit);
        rightStream[i] = tempBit;
        countBufferMessage++;
      }
    }
          //у потоках тепер є введене повідомлення. Оновлюємо потоки
          //початкового WAV-файлу
    _file.UpdateStreams(leftStream, rightStream);
  }

  public string ExtractMessage()
  {
    if (bufferLength > channelLength)
      throw new Exception();

    //отримуємо канали з файлу WaveAudio
    List<short> leftStream = _file.GetLeftStream();
```

```
List<short> rightStream = _file.GetRightStream();

            //витягуємо повідомлення з потоків і відображуємо у
            //відповідному полі

        //індекс у повідомленні, що шифрується
        int bufferIndex = 0;
        //ціла частина довжини повідомлення, що шифрується
        int messageLengthQuotient = leftStream[0];
        //дробова частина довжини повідомлення, що шифрується
        int messageLengthRemainder = rightStream[0];
        //довжина аудіоканалів
        int channelLength = leftStream.Count;

            //обчислюємо початкову довжину повідомлення, що шифрується
            int bufferLength = 32767 * messageLengthQuotient + messageLengthRemainder;
            //блок збереження повідомлення. Це значення дорівнює 1. Якщо
            //воно не дорівнює 1, то довжина повідомлення, що шифрується,
            //більше, ніж довжина початкового аудіо-файлу
            int storageBlock = (int)Math.Ceiling((double)bufferLength / (channelLength * 2));

        //створюємо повідомлення - масив байт
        byte[] bufferMessage = new byte[bufferLength];
        //йдемо по довжині потоку, починаючи з 1, тому що у [0]
            //зберігається довжина повідомлення; зберігаємо біти повідомлення
            //у лівий і правий потоки
        for (int i = 1; i < leftStream.Count; i++)
        {
            //беремо залишок від ділення на 8, тому що працюємо з
            //бітами; також працюємо з цифрою 7, тому що діапазон [0..7]
        if (bufferIndex < bufferLength && i % 8 > 7 - storageBlock && i % 8 <= 7)
           {
              //отримуємо біти повідомлення з лівого та правого каналів
              bufferMessage[bufferIndex++] = (byte)leftStream[i];
              if (bufferIndex < bufferLength)
                 bufferMessage[bufferIndex++] = (byte)rightStream[i];
           }
        }
        //перетворюємо масив байтів у рядкове повідомлення
            //і повертаємо його
        return Encoding.UTF8.GetString(bufferMessage);
    }
  }
}
```