Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Спеціальність 125 «Кібербезпека»

Освітня програма «Кібербезпека»

«Допущено до захисту»
В. о. завідувача кафедрою БІСТ
Мелкозьорова О. М.

_____

« » 2024 р.

Пояснювальна записка

до кваліфікаційної роботи бакалавра

на тему: «Методика адаптивного вбудовування цифрових водяних знаків в аудіофайли»

оцінка « »

Голова ЕК

Лемешко О. В. _____

Керівник к.т.н. Нарєжній О.П.

Рецензент ст. викл. Осипчук А.В.

Виконавець: студент групи КБ-42

Сорочинський С.І.

Харків – 2024

РЕФЕРАТ

Пояснювальна записка містить 56 сторінок, 14 рисунків, 7 таблиць, 1 додаток, 11 джерел.

Метою дипломної роботи є розробка та впровадження методики адаптивного вбудовування цифрових водяних знаків в аудіофайли.

Об’єктом дослідження дипломної роботи є стеганографія аудіофайлів, що використовуються в GSM мережах, для забезпечення автентифікації обох сторін встановленого зв’язку. Робота також спрямована на вивчення існуючих методик вбудовування цифрових водяних знаків в аудіофайли та використання при цьому різноманітних методів прямого розширення спектру сигналу. Для утворення основної досліджуваної методики використовуються послідовності, або коди Голда, адже їх використання утворює квазіортогональний сигнал, що дає змогу однозначно ідентифікувати користувачів під час встановлення зв’язку в мережі та відповідно ускладнює можливість компрометації вбудованих водяних знаків.

Предметом розробки є утворення послідовностей Голда та їх детальний аналіз, з ціллю довести їх стійкість щодо різноманітних атак зі сторони зловмисників, що надасть змогу повноцінно оцінити можливість використання даних послідовностей для налагодження однозначної ідентифікації абонентів, що в епоху стрімкого розвитку штучного інтелекту та відповідно створення великої кількості недостовірної інформації з використанням біометричних особливостей людини, що не дає однозначно ідентифікувати справжність інформації, але з використанням вбудованих водяних знаків автоматично визначається достовірність отриманого сигналу, у випадку даної роботи саме при використанні аудіофайлів.

Для досягнення поставленої мети та вирішення завдань в дипломній роботі було проаналізовано різноманітні методи розширення спектру та визначення їх особливостей та недоліків відносно один одного. Головним методом визначення достовірності інформації було використання та аналіз науково-технічної літератури з різних джерел, збір загальної інформації та висвітлення

найважливіших моментів, що мають вирішальне значення під час вибору основного напрямку дослідження.

Результатами проведеної роботи є практична реалізація вбудовування цифрового водяного знаку в аудіофайл на основі регістру зсуву з лінійним зворотним зв’язком з генерацією та використанням послідовностей Голда. Для цього було створено програмний код на мові програмування Java. Проведено детальний аналіз утвореного аудіосигналу з наявністю водяного знаку та без нього з обгрунтуванням доцільності використання даної методики в реально існуючих пристроях.

Ключові слова: ЦИФРОВИЙ ВОДЯНИЙ ЗНАК, КОДИ ГОЛДА, МЕТОД ПРЯМОГО РОЗШИРЕННЯ СПЕКТРУ, GSM, LFSR.

ABSTRACT

The explanatory note contains 56 pages, 14 figures, 7 tables, 1 appendix, and 11 sources.

The purpose of the thesis is to develop and implement a methodology for adaptive embedding of digital watermarks in audio files.

The object of research of the thesis is steganography of audio files used in GSM networks to ensure authentication of both parties of the established communication. The work is also aimed at studying the existing methods of embedding digital watermarks in audio files and using various methods of direct signal spectrum expansion. The main methodology under study is based on sequences or Gold codes, since their use creates a quasi-orthogonal signal that allows unambiguous identification of users during network communication and, accordingly, complicates the possibility of copromotion of embedded watermarks.

The subject of the development is the formation of Gold sequences and their detailed analysis in order to prove their resistance to various attacks by intruders, which will allow to fully assess the possibility of using these sequences to establish unambiguous identification of subscribers, which in the era of rapid development of artificial intelligence and, accordingly, the creation of a large amount of unreliable information using human biometric characteristics, which does not allow to unambiguously identify the authenticity of information, but using embedded watermarks.

To achieve this goal and solve the tasks, the thesis analyzes various methods of expanding the spectrum and identifying their features and disadvantages relative to each other. The main method of determining the reliability of information was the use and analysis of scientific and technical literature from various sources, collecting general information and highlighting the most important points that are crucial when choosing the main research area.

The results of this work are the practical implementation of embedding a digital watermark in an audio file based on a shift register with linear feedback with the generation and use of Gold sequences. For this purpose, a program code in the Java

programming language was created. A detailed analysis of the generated audio signal with and without a watermark is carried out with justification of the feasibility of using this technique in real-world devices.

Keywords: DIGITAL WATERMARK, GOLD CODES, DIRECT SPECTRUM EXPANSION METHOD, GSM, LFSR.

ЗМІСТ

ПЕРЕЛІК ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

GSM - Global System for Mobile Communications

TDMA - time division multiple access

FDMA - frequency division multiple access

TRX - transceivers

BCCH - broadcast control channel

DS-SS - Direct Sequence Spread Spectrum

FH-SS - Frequency Hopping Spread Spectrum

SNR - signal-to-noise ratio

CSS - Chirp Spread Spectrum

IEEE - Institute of Electrical and Electronics Engineers

BOK - binary orthogonal keying

DM - direct modulation

SSIS - Spread Spectrum Image Steganography

LSB - least significant bit

ДКП - дискретно-косинусне перетворення

LFSR - linear feedback shift register

ПВКФ - Періодична взаємно кореляційна функція

ЦВЗ – Цифровий водяний знак

ВСТУП

Сучасний розвиток цифрових технологій не лише сприяє зростанню обсягів цифрового контенту, але й породжує нові виклики в області забезпечення його автентичності та цілісності. Один із ключових аспектів цієї проблематики – захист цифрового контенту від несанкціонованого використання, копіювання або підробки. В цьому контексті виникає актуальна задача розробки ефективних методів захисту, серед яких особливе значення має адаптивне вбудовування цифрових водяних знаків в аудіофайли.

Досягнення у цій області включають в себе розвиток різноманітних методів вбудовування водяних знаків, вивчення їхньої стійкості до атак та оптимізацію їх використання для різних цілей. Проте, існують прогалини знань у питаннях ефективності та невидимості водяних знаків в аудіофайлах, а також у розробці адаптивних методик, що адаптуються до різних умов використання. Провідні фірми та вчені активно працюють над розвитком нових підходів у цій області. Їхні дослідження визначають світові тенденції вирішення поставлених задач і відображають перспективи подальшого розвитку.

Актуальність роботи полягає в необхідності розробки ефективних та надійних методів захисту цифрового аудіоконтенту, що враховують сучасні вимоги до безпеки та забезпечують високу стійкість до атак.

Метою даної роботи є розробка методики адаптивного вбудовування цифрових водяних знаків в аудіофайли, що забезпечує високу стійкість до атак та оптимальні характеристики водяних знаків, з урахуванням різноманітних умов використання.

Взаємозв'язок даної роботи з іншими науковими дослідженнями полягає в продовженні та розвитку існуючих підходів до вбудовування водяних знаків та внесенні нових методичних внесків у цю область. Результати даної роботи можуть бути корисними для досліджень у сферах цифрового захисту авторських прав, контролю за поширенням цифрового контенту та в управлінні правами на цифровий контент.

1 АНАЛІЗ МЕТОДІВ ПРЯМОГО РОЗШИРЕННЯ СПЕКТРУ

Розширення спектру (Spread Spectrum) - це технологія передачі сигналу, в якій ширина смуги заняття частоти значно перевищує швидкість даних. При модуляції з розширенням спектру, смуга пропускання сигналу, що передається, набагато більша, ніж смуга пропускання вихідного повідомлення, і визначається кодом розширення спектру, що в свою чергу є цифровим сигналом, який не залежить від повідомлення і відомий як передавачу, так і одержувачу. Модуляції з розширенням спектру забезпечують високий захист від завад, ускладнюють несанкціонований прийом повідомлень, дозволяють формувати кодовий мультиплекс, в якому всі користувачі працюють одночасно в одній і тій же смузі частот, і дають можливість вибіркової адресації окремих користувачів.

Перед тим як досліджувати основну тему роботи, слід проаналізувати існуючі методи прямого розширення спектру. Серед досліджуваних методів наявні наступні:

Метод розширення спектру із псевдовипадковою перебудовою частоти, або frequency hopping spread spectrum (FH-SS).

Розширення спектру методом прямої послідовності, або Direct Sequence Spread Spectrum(DS-SS).

Розширення спектру методом лінійної частотної модуляції, або Chirp Spread Spectrum(CSS).

1.1 Метод розширення спектру із псевдовипадковою перебудовою частоти

Метод розширення спектру із псевдовипадковою перебудовою частоти, або frequency hopping spread spectrum (FH-SS) передбачає, що несуча несуча, згенерована синтезатором, перестрибує з частоти на частоту в широкій смузі частот відповідно до псевдошумової кодової послідовності, визначеної генератором кодової послідовності[3]. Подібний модулятор зображено на рисунку 1.1.

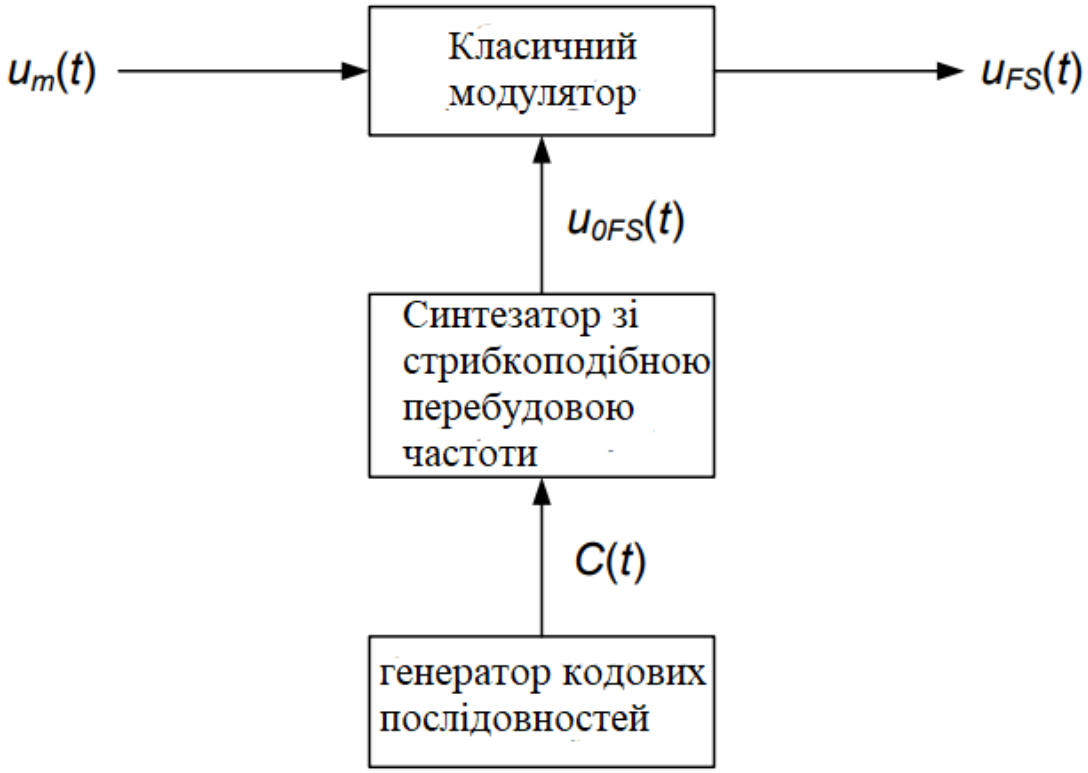

Рисунок 1.1 - Модулятор сигналу зі стрибкоподібною перебудовою частоти з широкосмуговим розширенням спектру

Модулятор сигналу FH-SS складається з трьох основних компонентів: генератора кодових послідовностей, синтезатора частоти і модулятора, в якому використовується одна з класичних аналогових або цифрових модуляцій. Ключовим елементом модулятора сигналів FH-SS є синтезатор частоти. Під керівництвом генератора кодових послідовностей на виході формується сигнал $u_{0FS}(t)$, який має змінну несучу частоту. Швидкість зміни несучої частоти може коливатися від декількох стрибків за секунду до 100000 стрибків за секунду[1]. Синтезатор частоти створює схему стрибкоподібної зміни частоти, яка є відомою як для передавача, так і для приймача. При використанні FH-SS ширина смуги частот для передачі сигналу значно перевищує ту, що потрібна для передачі повідомлення за допомогою деяких класичних модуляцій з несучою на фіксованій частоті. Коефіцієнт розповсюдження модуляцій FH-SS визначається таким чином:

$$\eta = \frac{B_{FS}}{B_m} \tag{1.1}$$

де B_{FS} позначає смугу частот передачі FH-SS, тоді як смуга частот одного каналу позначається B_m[1].

Якщо між сусідніми каналами немає частотного рознесення, то коефіцієнт рознесення дорівнює кількості каналів:

$$\eta = N \tag{1.2}$$

Якщо в будь-якому з каналів у смузі частот BFS присутній інший сигнал, цей сигнал і сигнал FH-SS створюють інтерференцію один з одним. Ця інтерференція має дуже коротку тривалість і виникає в інтервалі часу, коли сигнал FH-SS присутній в цьому каналі[3]. На рисунку 1.2 зображено класичний демодулятор стрибкоподібною перебудовою частоти з широкосмуговим розширенням спектру

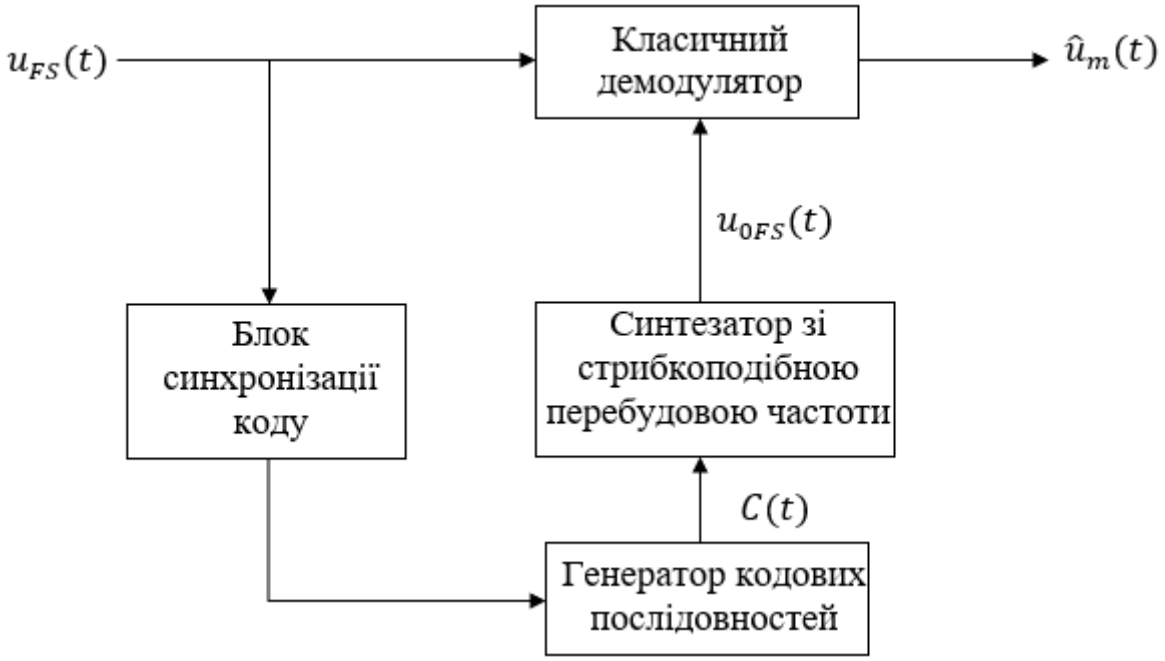

Рисунок 1.2 - Демодулятор сигналу зі стрибкоподібною перебудовою частоти з широкосмуговим розширенням спектру

Демодулятор сигналу FH-SS містить компоненти, такі як генератор кодових послідовностей, синтезатор зі стрибкоподібною перебудовою частоти, демодулятор і блок синхронізації коду. Основна мета приймача FH-SS полягає в розкладанні спектру вхідного сигналу, що передбачає виконання процесу розкладання перед демодуляцією. Це досягається шляхом порівняння вхідного сигналу з синхронізованою кодовою послідовністю, що локально генерується. Синхронізація коду є необхідним етапом для розширення спектру вхідного

сигналу. Блок кодової синхронізації відповідає за одночасну зміну несучої частоти в передавачі та приймачі. Системи FH-SS можуть бути швидкими або повільними. У швидких системах FH-SS один символ інформації передається за кілька частотних стрибків, тоді як у повільних системах FH-SS один або кілька символів інформації передаються в рамках одного частотного стрибка. Сигнали з розширеним спектром відзначаються високою стійкістю до ненавмисного або навмисного перешкоджання, що робить їх ідеальними для військових застосувань.

Технологія FH-SS володіє добрими характеристиками як у ближньому, так і у дальньому діапазонах і може бути реалізована легше, ніж розширення спектру методом прямої послідовності DS-SS. FH-SS демонструє високу стійкість до різних форм електронних атак і є ефективним засобом подолання перехоплення, пеленгування і глушіння. Технологія FH-SS широко використовується у військовій та комерційній сферах бездротового зв'язку, зокрема, для високо захищеної передачі даних. У військовій сфері FH-SS використовується для радіозв'язку на різних частотах та передачі сигналів управління і даних для безпілотних літальних апаратів. Військові радіостанції використовують технологію FH-SS для передачі повідомлень у аналоговому або цифровому форматі з різною швидкістю[1]. Ця технологія може бути реалізована за допомогою секретних ключів безпеки передач, обмін якими відбувається між передавачем і приймачем. FH-SS реагує на умови радіочастотного середовища, вибираючи найкращі частоти для роботи. Також можливе використання заборонених частотних діапазонів для роботи трансивера, що покращує ефективність зв'язку в умовах перешкоджання. Одним із застосувань методу FH-SS є технологія Global System for Mobile Communications, або GSM - це цифрова мобільна мережа, яка широко користується в Європі та інших частинах світу. GSM є стандартом для стільникових мереж другого покоління, тобто 2G. У системі GSM голосовий сигнал перетворюється на цифрові сигнали перед передачею через мережу. GSM працює у двох діапазонах частот: висхідний зв'язок відбувається у діапазоні від 890 МГц до 915 МГц, а низхідний - у діапазоні від 935 МГц до 960 МГц. Крім цих, є ще два діапазони у 1800 МГц: від 1710 МГц до 1785 МГц і від 1805 МГц до 1880 МГц. Система GSM використовує технології TDMA

та FDMA для оптимального використання частотного простору. Також, у системі використовується функція FH-SS для зменшення використання одних і тих же частот, що зменшує рівень перешкод. У системі GSM несучі частоти розділяються на вісім часових інтервалів згідно з TDMA, кожен з яких використовується для обробки виклику одного абонента. У випадку FH-SS, частоту можна змінювати кожні 4,615 мілісекунди, що відповідає довжині одного кадру TDMA, що дозволяє досягати 217 стрибків за секунду.

Базова станція GSM складається з однієї або декількох комірок, кожна з яких містить один або декілька прийомопередавачів. Перший часовий інтервал у першому приймачі використовується як канал управління широкомовленням (BCCH), а інші часові інтервали - як канали трафіку. У базовому діапазоні FH-SS, трансивери мають фіксовані частоти, але здійснюється стрибкоподібна зміна частоти, що дозволяє уникнути використання одних і тих же частот у багатьох трансиверах одночасно. Частоти перестрибують згідно з встановленою послідовністю, що залежить від кількості трансиверів. У системі FH-SS для базового діапазону передбачено дві групи стрибкоподібного переходу: одна для перших часових інтервалів кожного трансивера, окрім каналу трансляцією, або broadcast control channel (BCCH) і інша для всіх інших часових інтервалів. Режим стрибкоподібної зміни частот може бути випадковим або циклічним, залежно від вибраної схеми. Технологія Bluetooth - це стандарт для радіозв'язку короткого радіусу дії між персональними комп'ютерами, мобільними телефонами та іншими портативними пристроями, та власне є технологією що використовує метод FH-SS. Bluetooth стандартизується в рамках робочої групи або Institute of Electrical and Electronics Engineers (IEEE) 802.15 для бездротових персональних мереж. Стандарт IEEE 802.15.1 визначає основи бездротової технології Bluetooth. Технологія Bluetooth підтримує з'єднання типу «точка-точка» і «точка-багато точок». Піконет (piconet) - це основний термін, що використовується в технології Bluetooth для опису мережі, яка складається з одного майстра (master) та одного або декількох рабів (slave). Кілька піконет можуть бути створені і з'єднані між собою, де кожен піконет ідентифікується різною послідовністю FH-SS. Всі користувачі, які беруть

участь в одному піконеті, синхронізуються з цією послідовністю FH-SS. Піконет підтримує до 8 пристроїв, один з яких виступає в ролі ведучого. Майстер контролює трафік максимум до 7 пристроїв, які визначаються як підлеглі в піконеті. Усередині піконети Bluetooth-станції можуть встановлювати до трьох синхронних голосових каналів зі швидкістю 64 Кбіт/с або асинхронний канал передачі даних, що підтримує швидкість передачі даних до 723 Кбіт/с асиметрично або 433 Кбіт/с симетрично. Швидкість зміни частоти радіоканалу Bluetooth становить близько 1600 стрибків на секунду для передачі даних, або голосу і 3200 стрибків на секунду під час сканування сторінок і запитів[3]. Канал використовується протягом дуже короткого періоду, після чого відбувається перехід на інший канал, позначений заздалегідь визначеною псевдовипадковою послідовністю. Цей процес повторюється безперервно відповідно до послідовності FH-SS.

Bluetooth також забезпечує управління потужністю радіоканалу, де пристрої можуть домовлятися і регулювати свою радіопотужність відповідно до вимірювань рівня сигналу. Кожен пристрій у мережі Bluetooth може визначити рівень сигналу, який він отримує, і запросити інший пристрій мережі відрегулювати свій відносний рівень радіопотужності. Це робиться для економії енергії та утримання характеристик прийнятого сигналу в межах бажаного діапазону. Потужність передачі - від 1 мВт до 100 мВт. Розрахунковий робочий діапазон - від 1 до 100 м[3].

1.2 Розширення спектру методом прямої послідовності

Розширення спектру методом прямої послідовності, або Direct Sequence Spread Spectrum (DS-SS) - це технологія передачі, що передбачає поширення сигналу в ширшій смузі частот, ніж потрібно для простої передачі. Оригінальний сигнал даних комбінується з більш швидкісною послідовністю бітів, також відомою як чіп-код, для збільшення пропускної здатності сигналу. У DS-SS передавач приймає оригінальний сигнал даних і помножує його на шифрувальний код, в результаті чого отримується сигнал з більш широким частотним спектром. Цей код, або псевдовипадковий код, являє собою послідовність «0» та «1» на

набагато вищій частоті, ніж оригінальний сигнал[4]. Приймач, використовуючи той самий код, може потім розділити поширений сигнал і відновити оригінальні дані. Це означає, що навіть якщо частина сигналу буде пошкоджена під час передачі, приймач все одно зможе відновити оригінальні дані, тим самим зменшуючи ймовірність втрати або пошкодження даних. Цей процес поширення і рознесення сигналу забезпечує стійкість DS-SS до перешкод і здатність підтримувати цілісність сигналу.

Розширений спектр - це засіб передачі, в якому сигнал займає смугу частот, що перевищує необхідну для передачі інформації. Сигнал займає смугу частот, що перевищує мінімально необхідну для передачі інформації. Розширення смуги досягається за допомогою коду, який не залежить від даних, а синхронізований прийом з кодом на приймачі використовується для ущільнення і подальшого відновлення даних. Аналогічно, розширення спектру чирп-діапазону отримало свою назву завдяки використанню модульованих чирпів для передачі і пов'язаних з ними методів імпульсного стиснення для кодування інформації. Концепції узгоджених фільтрів та імпульсного стиснення є фундаментальними для системи CSS. Згідно з теорією імпульсного стиснення, імпульсне стиснення чирп-сигналу може бути реалізоване за допомогою узгодженого фільтра, який є оптимальним фільтром з точки зору досягнення максимального відношення сигнал/шум, або signal-to-noise ratio (SNR) на виході фільтра для виявлення сигналів у середовищі білого гауссівського шуму.

Структура системи розширення спектру прямої послідовності зображена на рисунку 1.3 та на рисунку 1.4[1]:

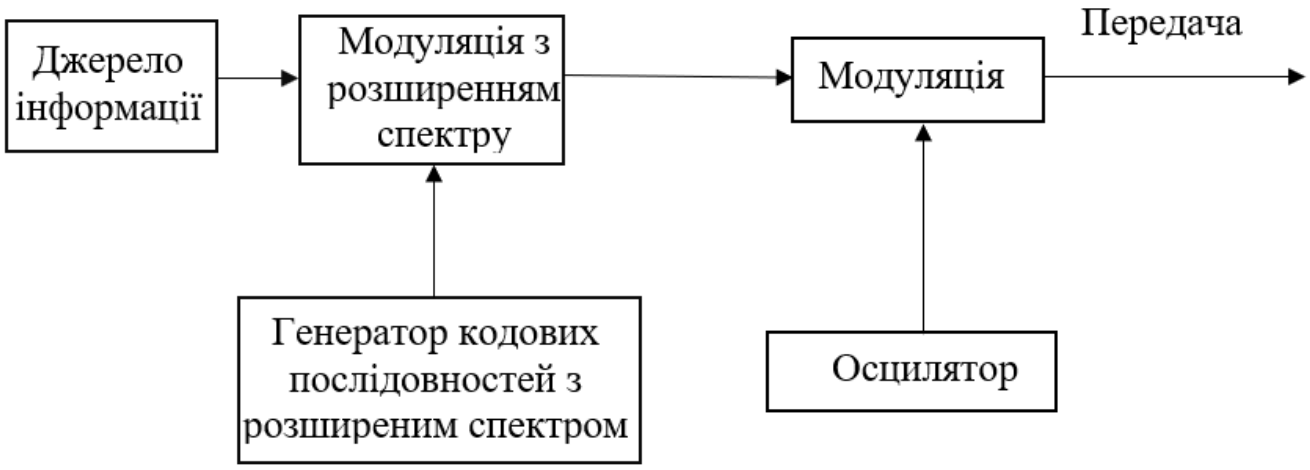

Рисунок 1.3 – Передача сигналу в системі розширення спектру прямої послідовності

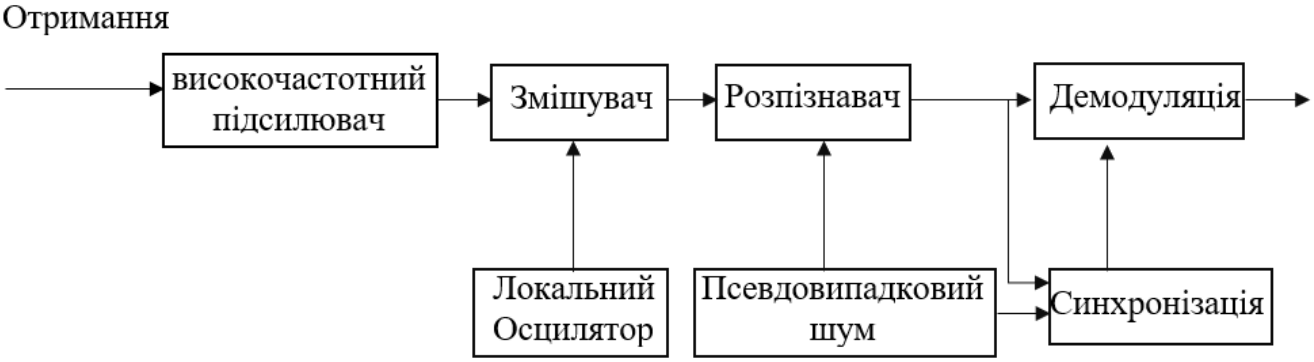

Рисунок 1.4 – Отримання сигналу в системі розширення спектру прямої послідовності

На передавачі інформація базової смуги спочатку модулюється в цифровий сигнал, тобто двійкову послідовність цифр[1]. Частотний спектр вихідного сигналу може бути розширений додаванням за модулем два послідовності коду з високою швидкістю розширення спектру та послідовності інформаційного коду. Розширений сигнал буде модульований на радіочастоту і відправлений. Як правило, в системі DS-SS використовується фазова маніпуляція, тоді як частотна маніпуляція, або амплітудна маніпуляція використовуються рідко. На приймачі прийнятий радіочастотний сигнал спочатку стає сигналом проміжної частоти після частоти змішування, а потім використовується код синхронного розширення

спектру, який розширює спектр сигналу. Нарешті, вихідний інформаційний сигнал може бути отриманий шляхом демодуляції.

Сигнал з розширеним спектром буде перетворений на радіочастотний сигнал та висланий. Для цього процесу використовується фазова модуляція, оскільки такий сигнал має велику стійкість до перешкод і вищий рівень безпеки порівняно з частотною модуляцією. Один з методів - це двійкова фазова модуляція, де початкова фаза «0» відповідає цифрі «0», а «π» - цифрі «1». Комбінований код, який складається з інформаційної послідовності та коду розширеного спектру, безпосередньо впливає на несучу хвилю. Однак цей метод менш ефективний, ніж квадратурна фазова модуляція, оскільки має меншу пропускну здатність при однаковій потужності передачі та швидкості передачі даних. Крім того, він менш стійкий до перешкод порівняно з двійковою модуляцією. Ймовірність виявлення сигналу під час передачі обернено пропорційна ширині смуги сигналу і прямо пропорційна відношенню енергії сигналу до шуму. У системі DS-SS спектр вихідного сигналу розширюється, що призводить до широкої смуги пропускання і низької потужності на одиницю смуги пропускання. Таким чином, спектральна щільність потужності сигналу з розширеним спектром нижча, ніж у шумі, що робить сигнал менш помітним у шумі. Це ускладнює перехоплення та виявлення корисного сигналу зловмисником. Система DS-SS вимагає широкої смуги частот.

Якщо б існувала тільки одна адреса для передачі даних, використання частотного спектру було б обмеженим. Тому зв'язок з декількома адресами в одній широкій смузі частот може покращити використання частотного спектру. У системі DS-SS лише одержувач, який має відповідний код розширення спектру, може розкодувати сигнал для отримання вихідного сигналу. Відправник використовує різні коди розширення спектру для розширення вихідного сигналу і надсилає їх різним одержувачам. Приймачі використовують відповідні коди для отримання потрібної інформації. Таким чином, всі приймачі можуть спілкуватися з відправником в одній і тій же смузі частот, не заважаючи один одному[4].

Існує багато різних видів завад, таких як білий шум, одночастотні завади, вузькосмугові завади, багатопроменеві завади та імпульсні завади тощо. Загалом,

потужність вузькосмугових завад є високою, тоді як їхня смуга пропускання є малою. Система DS-SS має відмінну стійкість до вузькосмугових завад. Припустимо, що під час передачі корисного інформаційного сигналу виникає вузькосмуговий сигнал завади. На приймачі цей змішаний сигнал буде дешифрований за допомогою коду з синхронним розширенням спектру. Корисний інформаційний сигнал буде відновлений до вузькосмугового, в той час як сигнал завади буде поширений до широкосмугового. Таким чином, спектральна щільність потужності корисного сигналу збільшиться, в той час як потужність сигналу завади значно зменшиться. Потім ці змішані сигнали проходять через фільтр. Тільки та частина сигналу завади, яка знаходиться в тій самій смузі, може пройти через фільтр.

Слід провести порівняння з методом розширення спектру зі скачкоподібною перебудовою частоти FH-SS. Він в свою чергу використовує двійкову псевдовипадкову послідовність для керування частотою несучої хвилі, виробленої синтезатором частоти. Інформаційний сигнал модулює несучу хвилю і стає радіочастотним сигналом. Відомо, що DS-SS має хорошу завадостійкість, але якщо є сильна завада, яка має незмінну частоту, її продуктивність буде страждати від сильної завади, оскільки ця завада вже перевищила завадостійкість системи. Однак для FH-SS завадою може бути лише та завада, частота якої збігається з частотою FH-сигналу. Таким чином, FH-SS має кращу здатність протистояти суцільним частотним завадам. Для безпеки зв'язку миттєва щільність спектру потужності FH-SS є високою, тоді як щільність спектру потужності DS-SS є нижчою. Сигнал DS-SS повністю тоне в шумі, тому його складніше виявити. Для ефекту ближнього і дальнього радіусу дії система DS-SS зазнає значного впливу, тоді як система FH-SS меншого[4]. Через втрати на шляху проходження потужність сигналу, виробленого ближнім передавачем, набагато більша, ніж потужність сигналу, відправленого віддаленим передавачем. Однак у системі FH-SS несуча частота контролюється псевдовипадковим кодом і стрибає за певними правилами. Таким чином, зв'язок не відбувається на фіксованій частоті. Ці два передавачі можуть працювати на різних частотах, щоб уникнути ефекту ближнього і дальнього зв'язку.

Що стосується багатопроменевих завад, то якщо багатопроменеве часове реле перевищує ширину одного псевдокоду, то багатопроменеві завади мало впливають на сигнал DS-SS, тому що їх можна розглядати як шум і усунути. Але для сигналу FH-SS єдиним способом протистояти багатопроменевій заваді є збільшення швидкості перестроювання частоти, але цей метод підвищує вимоги до системи FH-SS. Таким чином, система DS-SS має кращу здатність протистояти багатопроменевим завадам.

Можна зробити висновок, що хоча система DS-SS має багато переваг, вона все ще має деякі недоліки. У реальних застосуваннях, для тих систем зв'язку, які вимагають високої продуктивності, зазвичай використовується гібридна технологія розширення спектру, така як FH/DS. Ця технологія гібридного спектру поєднує в собі переваги DS-SS і FH-SS і покращує недоліки кожної з них. Однак використання гібридної технології розширення спектру може ускладнити систему і підвищити її вартість. Тому в майбутньому складність системи повинна бути спрощена, а вартість знижена.

1.3 Розширення спектру методом лінійної частотної модуляції

Метод розширення спектра методом лінійної частотної модуляції, або Chirp Spread Spectrum (CSS).У CSS дані кодуються за допомогою чирплетів, і чирплет-сигналів може збільшуватися (up-chirp) або зменшуватися (down-chirp) в частоті з плином часу. Чирплет (chirp) - це основна одиниця чирплет-сигналу, або деяка кількість даних, збережених в термінах частоти та часу. Ця одиниця містить інформацію про те, як частота сигналу змінюється з часом. У методі CSS, чирплети використовуються для кодування і передачі даних через спектр сигналу, що дозволяє ефективно використовувати спектральні ресурси і забезпечувати високу стійкість до перешкод. Чирплет-сигнал - це сигнал, частота якого лінійно змінюється з часом. У CSS чирплет-сигнали використовуються для кодування і передачі даних, що дозволяє досягти високої стійкості до перешкод та більшої ефективності використання спектра.

Розрізняється два основних типи методу CSS: Бінарна ортогональна маніпуляція, або Binary orthogonal keying (BOK) і пряма модуляція, або Direct modulation (DM)[5].

На відміну від методів FH-SS і DS-SS, які використовують псевдовипадкове кодування для розширення спектра інформаційного сигналу, CSS не вимагає додаткового кодування для розширення спектра, оскільки може икористовувати для кодування чирп-сигнал. Крім того, сигнали CSS демонструють вищий ступінь завадостійкісті, ніж чистий синусоїдальний сигнал, що робить цей клас сигналів хорошим кандидатом для використання в системах зв'язку з розширеним спектром. Метод CSS також виявився стійким до допплерівських та інших спотворюючих ефектів. Бездротові системи на основі CSS можуть передавати сигнал при низькій потужності передачі за рахунок розширення спектру[5]. Тому CSS є гарним вибором для бездротового зв'язку завдяки своїм вродженим перевагам, таким як низька потужність передачі, простота реалізації та хороша здатність відкидати завади.

Спектр поширення чирпових сигналів використовує чирпи для сигналізації при передачі даних, а також використовує пов'язану з ним техніку імпульсного стиснення для декодування інформації. Чирп - це сигнал, в якому частота змінюється протягом певного часового інтервалу, а техніка імпульсного стиснення є практичною реалізацією узгодженої фільтрації. Незважаючи на те, що метод імпульсного стиснення застосовувався в радіолокаційних системах з початку п'ятдесятих років, перша відома робота, що пропонує її для інших застосувань, окрім радіолокації, була опублікована лише в 1962 році. Розширений спектр чирпів, що використовувався в радіолокаційних системах в минулому, привертає все більше і більше уваги для низькошвидкісних бездротових персональних мереж LR-WPAN. Спектральна модуляція з розширенням спектру була поширена як метод передачі даних через внутрішні канали в щільних багатопроменевих середовищах. У березні 2007 року Інститут інженерів з електротехніки та інженерів з електроніки (IEEE), прийняв CSS як методи, що є основою фізичного рівня у новому бездротовому стандарті IEEE 802.15.4a. Це заклало основу для широкого

розповсюдження CSS в різних додатках, таких як системи визначення місцезнаходження в реальному часі, промисловий контроль і сенсорні мережі[5]. На рисунку 1.5 зображена Блок-схема системи BOK CSS.

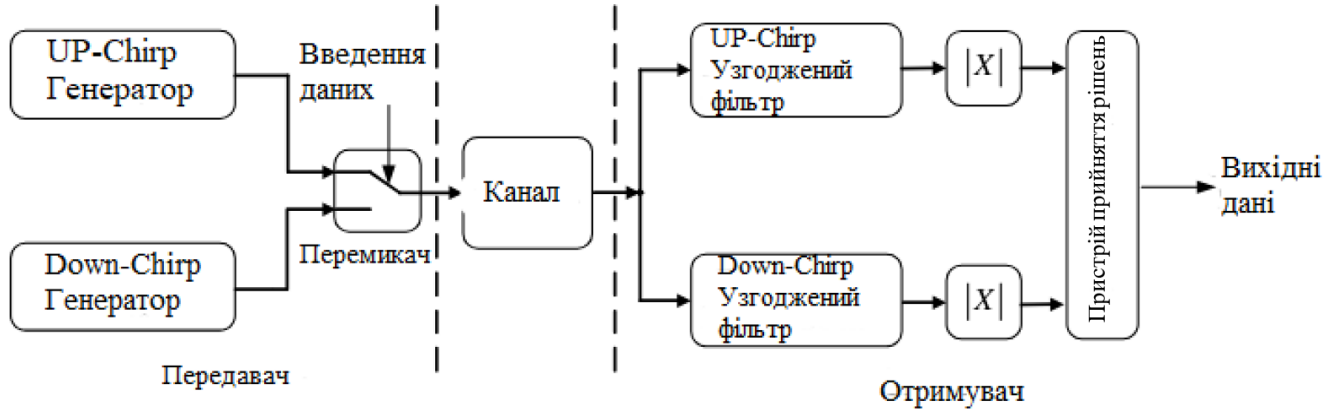

Рисунок 1.5 - Блок-схема системи BOK CSS

Система BOK CSS використовує два різних чирпи з однаковою смугою пропускання і тривалістю, але протилежною полярністю розгортки, наприклад, лінійний висхідний чирпи і лінійний низхідний чирпи. Тут висхідні та низхідні сигнали використовуються для представлення різних символів даних. Наприклад біти «1» і «0» можуть бути представлені чирпами з позитивною і негативною миттєвою швидкістю зміни частоти відповідно. На стороні приймача відповідні фільтри використовуються для декодування отриманого сигналу[5]. На рисунку 1.6 зображена блок-схема системи DM CSS з модулятором/демодулятором з квадратурною фазовою маніпуляцією.

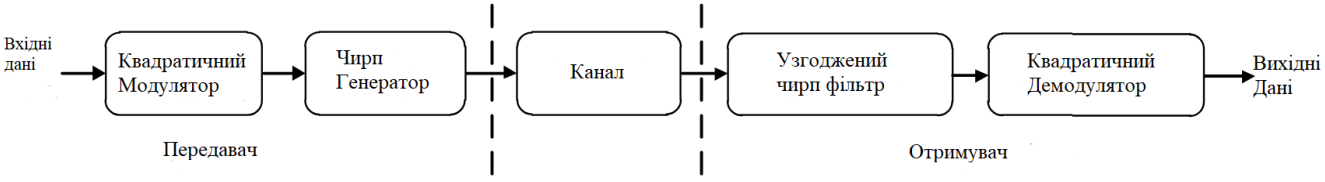

Рисунок 1.6 - Блок-схема системи DM CSS з модулятором/демодулятором з квадратурною фазовою маніпуляцією

Система DM CSS використовує звукові сигнали як механізм розподілу часу, а не сигналізації. Ця система схожа за концепцією на системи DS-SS. Подібно до псевдошумової послідовності в системах DS-SS, чирп-сигнал виконує аналогічну функцію в системі DM CSS. У системах DM CSS чирпи використовуються лише з

метою розповсюдження і де-розповсюдження, тоді як дані модулюються за допомогою звичайної некогерентної модуляції. Тому система DM CSS складніша в реалізації порівняно з системою BOK CSS. Метод DM повинен поєднуватися з іншими схемами цифрової модуляції для передачі даних, в той час як метод BOK передачі даних не має такої можливості[5]. Таким чином, схема модуляції методу BOK для бездротової передачі даних є простішою, ніж схема модуляції методу DM. Крім того, метод BOK використовує два різних сигнали для модуляції даних, в той час як DM метод може використовувати для поширення лише один чирп-сигнал. Отже, метод DM фокусується на автокореляції чирп-сигналу, яка є такою ж, як у радіолокаційній системі, в той час як метод BOK залежить не тільки від автокореляційних характеристик, але й від крос-кореляційних характеристик між двома обраними чирпами. Метод BOK більш підходить для аналізу чирпів, оскільки його продуктивність більше залежить від характеристик чирпів, ніж метод DM.

2 СТЕГАНОГРАФІЯ З РОЗШИРЕНИМ СПЕКТРОМ. КОДИ ГОЛДА ТА ПОСЛІДОВНОСТІ КАСАМІ

2.1 Концепція стеганографії з розширеним спектром

Концепція стеганографії з розширеним спектром, або Spread Spectrum Image Steganography (SSIS), яка використовує завадостійкі властивості технології DS-SS з метою приховування даних у зображеннях. Ця концепція пропонує ряд покращень та змін у порівнянні з попередніми підходами. Встановлено, що методи SSIS дуже стійкі до широкого спектру операцій обробки зображень, таких як стиснення, розмиття та зміна розміру. Ця стійкість робить SSIS цінним інструментом для безпечного зв'язку, оскільки вона гарантує, що вбудовані дані залишаються захищеними, навіть при суттєвих змінах зображення. Розвиток методів SSIS, які можуть протистояти виявленню як статистичними, так і методами стеганоаналізу на основі машинного навчання, є значним досягненням в цій області[6]. Ці досягнення сприяли підвищенню надійності та безпеки SSIS, навіть за наявності передових методів стеганоаналізу. SSIS - це інноваційна технологія, яка поєднує в собі принципи стеганографії та розглянутої раніше технології DS-SS для приховування інформації всередині зображень. SSIS використовує завадостійкі властивості DS-SS для забезпечення надійного та безпечного методу приховування даних у зображеннях прикриття. Основною характеристикою SSIS є використання технології DS-SS, яка поширює сигнал у широкій частотній смузі за допомогою коду, що розноситься. В контексті SSIS, прихована інформація модулюється за допомогою розширювального коду, а зображення прикриття розглядається як шум у каналі зв'язку. Такий підхід дозволяє SSIS скористатися перевагами завадостійкості і конфіденційності DS-SS, що призводить до створення високонадійної та надійної стеганографічної системи[6].

У порівнянні з іншими стеганографічними методами, SSIS виділяється виділяється своєю стійкістю, безпекою та збереженням якості зображення прикриття. Традиційні методи, такі як вбудовування найменш значущих бітів, або least significant bit (LSB) та дискретно-косинусне перетворення (ДКП), можуть

вносити помітні спотворення зображення на обкладинці або бути вразливими до стеганоаналізу. На відміну від них, SSIS усуває ці обмеження, використовуючи завадостійкі властивості DS-SS-технології та адаптивних послідовностей розкладання. Теоретичні основи технології прямого розширення спектра (DS-SS) ґрунтуються на фундаментальній теоремі Шеннона-Хартлі, яка визначає максимальну швидкість, з якою інформація може бути передана по каналу зв'язку для певної смуги пропускання в присутності шуму[6].

Математично теорема Шеннона-Хартлі має вигляд

$$C = \Delta F log_2(1 + \frac{P_s}{P_N}) \tag{2.1}$$

де C - пропускна здатність каналу в бітах за секунду (біт/с), ΔF - ширина смуги частот в герцах(Гц), P_s - потужність корисного сигналу, P_N - потужність адитивного білого гаусівського шуму, а $\frac{P_s}{P_N}$ відношення сигнал/шум, або signal-to-noise ratio (SNR).

Формула підкреслює прямий зв'язок між пропускною спроможністю каналу і пропускною здатністю та SNR, що підтверджує актуальність технології DS-SS. Використання технології DS-SS підвищує продуктивність зв'язку, особливо в умовах низького SNR, за рахунок використання довгих послідовностей з рознесенням для розширення частотного спектру сигналу, тим самим забезпечуючи високошвидкісну передачу даних і зменшує вразливість до завад та підслуховування.

Аналітичні вирази для функцій крос-кореляції та автокореляції наведені нижче[6].

$$R_{ij}(k) = \frac{1}{N}\sum_{n=0}^{N-1} c_i(n)c_j(n+k), R_{ii}(k) = \frac{1}{N}\sum_{n=0}^{N-1} c_i(n)c_j(n+k) \tag{2.2}$$

$R_{ij}(k)$ та $R_{ii}(k)$ представляють відповідно крос-кореляційну та автокореляційну функції, $c_i(n)$ та $c_j(n)$ позначають послідовності поширення для користувачів i та j, k - часовий зсув, а N - довжина послідовності поширення.

Суть методики стеганографії з розповсюдженим спектром (SSIS) полягає в адаптації принципів роботи системи розповсюдженого спектра (DS-SS) для

приховування інформації у цифрових зображеннях. Підхід SSIS розглядає зображення як зашумлений канал зв'язку, де зашифровані дані передаються шляхом розповсюдження в обрану послідовність, яка потім вбудовується у зображення з мінімальними візуальними спотвореннями. Цей метод дозволяє ефективно приховувати інформацію в цифрових медіафайлах, забезпечуючи високу стійкість та конфіденційність даних та визначається наступною формулою[6].

$$y(t) = s(t) + Gq(t) \tag{2.3}$$

де $y(t)$ - стего-зображення, $s(t)$ - сигнал зображення прикриття, $q(t) = x(t)c(n)$ - сигнал поширення секретного повідомлення $x(t)$ за допомогою послідовності $c(n)$, а G - сила вбудовування.

Видобування секретних даних вимагає знання послідовності розповсюдження та потужності вбудовування. Процес ґрунтується на співставленні отриманого стего-зображення з послідовністю поширення, щоб виокремити вбудований сигнал від зображення[6].

$$z(t) = \frac{1}{N}\sum_{n=0}^{N-1} y(t-n)c(n) = G\frac{1}{N}\sum_{n=0}^{N-1} q(t-n)c(n) + \frac{1}{N}\sum_{n=0}^{N-1} s(t-n)c(n) \tag{2.4}$$

де $z(t)$ - результат кореляції, що забезпечує основу для відновлення початкового секретного повідомлення.

2.2 Вдосконалення технології SSIS

Якщо розглянути формули згадані раніше, які описують процес модуляції даних користувача шляхом поширення послідовностей і відновлення даних шляхом обчислення кореляції. Використання великих ансамблів слабко корельованих послідовностей розкладу дає змогу досягти малих значень $R_{ij}(k)$ при $i \neq j$. Це означає, що при відновленні $x_i(t)$, перший доданок у правій частині частини формули (2.4) буде дорівнювати[7].

$$G\sum_{j=0}^{M-1} x_i(t)R_{ij}(k) \approx Gx_i(t) \tag{2.5}$$

Тобто, формулу відновлення даних можна записати як

$$m_i = sign\big(z(t)\big) = \begin{cases} -1, z(t) < 0 \\ +1, z(t) > 0 \end{cases} \quad (2.6)$$

Формула (2.5) добре працює, коли другий доданок у формулі (2.4) близький до нуля. Це припущення означає, що послідовності розсіювання не корелюють із зображенням контейнера. Наприклад, якщо всі послідовності розкидання задовольняють наступній умові, то можна досягти безпомилкового відновлення інформації[7].

$$\left| \frac{1}{N} \sum_{n=0}^{N-1} s(t-n) c_i(n) \right| < G_{max} \quad (2.7)$$

Перший значний внесок у розвиток технології SSIS полягає в адаптивній генерації послідовностей розкидання, що враховують статистичні властивості контейнерів. Цей підхід має на меті суттєво знизити інтенсивність помилок у відновлених даних і, за певних умов, навіть дозволяє відновлювати повідомлення без помилок. Другий внесок полягає у використанні нового методу приховування даних, що базується на прямій послідовній адресації[7].

Замість добутку $x(t)c(n)$ у формулі (2.3) використовується правило

$$Q(t) = c_x(n) \quad (2.8)$$

де X - адреса послідовності розсіювання, її номер у списку всіх використаних спредінг-кодів. Щоб проілюструвати це, слід розглянути простий приклад.

2.3 Коди Голда та послідовності Касамі

Коди Голда, названі на честь Роберта Голда, - це спеціальні типи двійкових послідовностей з бажаними властивостями для використання в телекомунікаційних і супутникових навігаційних системах. Вони характеризуються низькою взаємною перехресною кореляцією, що робить їх ідеальними для таких застосувань, як множинний доступ з кодовим поділом каналів (CDMA) і глобальна система позиціонування (GPS). Коди Голда, що використовуються в системах з кодовим поділом каналів, створюються шляхом комбінування пар лінійних кодових послідовностей з псевдовипадковим шумом максимальної довжини.

Характеристики цих кодів визначаються конкретними обраними послідовностями максимальної довжини. Існує важлива відмінність між послідовностями та формами сигналів. Форми сигналів - це аналогові сигнали з такими характеристиками, як амплітуда, фаза і частота. Послідовності, з іншого боку, є оцифрованими послідовностями одиниць і нулів, де кожен символ представляє певний логічний стан форми сигналу протягом певного часового інтервалу. окремі одиниці і нулі в кодовій послідовності називаються чипами, або chip[8]. У широкосмуговому зв'язку цей часовий інтервал відомий як час чипа. Швидкість, з якою генератор коду видає ці послідовності, називається швидкістю передачі.

Послідовності зсувних регістрів - це кодові послідовності, що генеруються зсувними регістрами з логікою зворотного зв'язку, яка повертає логічні комбінації станів зсувного регістра на свій вхід. Ці послідовності є циклічними, тобто вони повторюються безперервно, доки каскади зсувного регістра перебувають в тактовому режимі. Довжина послідовності - це кількість мікросхем у кожному повторюваному циклі послідовності, а частота мікросхем - це тактова частота зсувного регістра. Послідовності зсувного регістра можуть мати перехідні процеси, нециклічні послідовності одиниць і нулів, що генеруються перед початком циклічної послідовності. Зсувний регістр генерує лінійну послідовність, якщо його функція зворотного зв'язку може бути виражена як сума за модулем 2[8].

У зв'язку з розширеним спектром сигналу форма сигналу, що використовується для розширення спектра сигналу, повинна бути передбачуваною, щоб модульований сигнал міг бути успішно відновлений в приймачі. Водночас, для досягнення бажаного ефекту ця форма сигналу повинна бути схожою на випадковий шум. Щоб досягти такої псевдовипадкової шумоподібної поведінки, кодова послідовність, що лежить в основі, повинна відповідати трьом постулатам випадковості.

Перший постулат - це властивість балансу, яка вимагає, щоб кількість одиниць і нулів у кодовій послідовності була майже однаковою при будь-якому зсуві фази послідовності. Цей баланс важливий, оскільки він мінімізує постійну

складову в генерованому сигналі і модульованому сигналі, що має вирішальне значення в схемах модуляції, де потрібно придушити несучу складову[8].

Другий постулат випадковості фокусується на розподілі одиниць і нулів у послідовності по довжині відліків. Такт - це послідовна серія одиниць або нулів, а довжина такту - це кількість послідовних одиниць або нулів. Такий розподіл означає, що буде однакова кількість послідовностей одиниць і нулів однакової довжини. Якщо кодова послідовність відповідає певному розподілу довжини прогонів, згаданому раніше, то розподіл чипів у ній стає статистично незалежним. Ця незалежність ще більше посилює його випадкову природу.

Третій постулат випадковості стверджує, що автокореляційна функція псевдовипадкової кодової послідовності повинна бути двозначною. Автокореляція в цьому контексті передбачає порівняння відповідних фішок між оригінальною послідовністю та її фазово зсунутою версією[8]. Кожна пара чипів оцінюється як узгоджена, тобто обидва чипа дорівнюють або 1, або 0 та неузгоджена. Значення кореляції обчислюється як різниця між кількістю згодних і незгодних. Для виконання цього постулату значення автокореляції в ідеалі має бути набагато вищим при нульовому зсуві, тобто ідеальної кореляції, порівняно з іншими зсувами, де значення має бути близьким до нуля. Ця характеристика дуже нагадує автокореляційну функцію білого гаусівського шуму, яка є дельта-функцією на початку координат. Ця двозначна властивість автокореляції має важливе значення для синхронізації приймача в розширеному спектрі зв'язку. Приймач може легко і надійно синхронізуватися з вхідним сигналом, якщо він має єдиний, чіткий кореляційний пік при нульовому зсуві. Це спрощує конструкцію приймача і зменшує ймовірність помилкової синхронізації[9].

Зв'язки зворотного зв'язку в генераторі зсувного регістра визначають, чи є вихідна послідовність максимальною чи ні. Генератор, призначений для послідовностей максимальної довжини, вироблятиме лише одну таку послідовність, причому початкові умови визначають фазовий зсув на виході. Генератори не максимальної довжини, з іншого боку, можуть виробляти декілька послідовностей залежно від початкових умов. Послідовності максимальної

довжини мають близькі до ідеальних характеристики і широко використовуються в системах зв'язку і дальності завдяки дотриманню трьох постулатів випадковості. Вони мають збалансований розподіл одиниць і нулів, близький до ідеального розподіл довжини пробігу та двійкову автокореляційну функцію.

Послідовності максимальної довжини мають й інші помітні властивості:

Вони можуть бути згенеровані лише генераторами з регістром зсуву з одним зворотним зв'язком і парною кількістю відгалужень зворотного зв'язку.

Порядок виведення послідовності може бути змінено на протилежний, якщо змінити положення відгалужень зворотного зв'язку.

Вони мають три спеціальні властивості лінійного додавання[9].

Додавання послідовності максимальної довжини до будь-якої фазово зсунутої копії самої себе призводить до створення іншої фазово зсунутої копії оригіналу.

Додавання двох послідовностей максимальної довжини різної довжини дає відрізок довшої послідовності максимальної довжини, якщо вихідні послідовності вибрані правильно.

Додавання двох послідовностей максимальної довжини однакової довжини дає послідовність тієї ж довжини з бажаними властивостями авто- та крос-кореляції, за умови, що вихідні послідовності підібрані правильно.

Ця остання властивість особливо важлива для генерації кодів Голда, оскільки вона дозволяє створювати композитні послідовності з відмінними кореляційними властивостями шляхом об'єднання двох правильно підібраних послідовностей максимальної довжини.

Основною метою генерування послідовностей розсіювання є максимізація кардинальності M, набору послідовностей довжини N, для яких взаємна кореляція не перевищує ρ_{max}. Границя Велша є відомою фундаментальною межею яка з'єднує M, N та ρ_{max}, встановлюючи обмеження на квадрат кореляції, $(\rho_{\max})^2$, для різних послідовностей заданої довжини N та кардинальності M[8].

Нехай $\{x_1, \ldots, x_M\}$ - двійкові вектори довжини N (всі елементи кожного вектора x_i дорівнюють 1 або -1).

$$\rho_{max} = \frac{max_{i \neq j}\langle x_i, x_j \rangle}{N} \tag{2.9}$$

де $\langle x_i, x_j \rangle$ точковий добуток векторів x_i та x_j. Тоді для $k = 1, 2, \ldots$, виконується границя Велша[8].

$$(\rho_{max})^{2k} \geq \frac{1}{M-1}\left[\frac{M}{(N+k-1)} - 1\right] \tag{2.10}$$

Очевидно, що при $k = 1$ маємо границю на квадраті взаємної кореляції послідовностей x_i[8].

$$(\rho_{max})^2 \geq \frac{1}{M-1}\left[\frac{M}{N} - 1\right] = \frac{M-N}{(M-1)N} \tag{2.11}$$

Для асинхронних методів DS-SS всі послідовності x_i повинні бути статистично некорельованими для довільних випадкових початкових точок. Іншими словами, на додаток до кожної послідовності x_i ми також повинні розглядати всі циклічно зсунуті копії x_i. У цьому випадку, у формулі (2.11) слід замінити M на MN, що дасть результат[8].

$$(\rho_{max})^2 \geq \frac{MN-N}{(MN-1)N} = \frac{M-N}{MN-1} \tag{2.12}$$

Формула (2.12) встановлює фундаментальну межу, нижче якої квадрат взаємної кореляції між будь-якими циклічно зсунутими копіями різних послідовностей не може опуститися нижче, ніж квадрат взаємної кореляції між будь-якими циклічно зсунутими копіями. Для випадку, коли $M \gg 1$, формула (2.12) набуває простого вигляду[8].

$$(\rho_{max})^2 \geq \frac{1}{N} \tag{2.13}$$

Тоді як абсолютна взаємна кореляція генерованих сигналів часто порівнюють з величиною $\frac{1}{\sqrt{N}}$.

Коди Голда та послідовності Касамі - це дві родини послідовностей, що розповсюджуються, які привернули значну увагу завдяки своїм бажаним кореляційним властивостям[9]. Коди Голда утворюються шляхом комбінування двох послідовностей зсувного регістру максимальної послідовностей регістрів зсуву максимальної довжини (m-послідовностей), згенерованих лінійними

регістрами зсуву зі зворотним зв'язком, або linear feedback shift register (LFSR) з певними парами примітивних поліномів, яким надається перевага. Нехай $s_1(n)$ та $s_2(n)$ дві m-послідовності довжини $N = 2m - 1$, де m довжина LFSR. Код Голда отримується шляхом побітового додаванням цих m-послідовностей за модулем 2[9].

$$(n) = s_1(n) \oplus s_2(n), \text{ for } n = 0,1, \dots, N-1 \tag{2.14}$$

Набір з $M = N + 2$ послідовностей Голда можна згенерувати з пари m-послідовностей. Коди Голда мають тризначну функцію крос-кореляції, підставляючи максимальне у формулі (2.14), отримаємо максимальне абсолютне значення.

$$(\rho_{Goldmax})^2 = \left(\frac{1 + 2^{\frac{n+1}{2}}}{2^n - 1}\right)^2 = \left(\frac{\sqrt{2(N+1)} + 1}{N}\right)^2 \tag{2.15}$$

Наближення за допомогою ряду Пюізо дає

$$(\rho_{Goldmax})^2 \approx \frac{2}{N} + 2\sqrt{2}(\frac{1}{N})^{\frac{3}{2}} + \frac{3}{N^2} + \sqrt{2}(\frac{1}{N})^{\frac{5}{2}} - \frac{(\frac{1}{N})^{\frac{3}{2}}}{2\sqrt{2}} + \frac{(\frac{1}{N})^{\frac{9}{2}}}{4\sqrt{2}} + O((\frac{1}{N})^{\frac{11}{2}}) \tag{2.16}$$

Використовуючи лише перший член ряду, маємо $(\rho_{Goldmax})^2 \approx \frac{2}{N}$. Це означає, що для великих значень N крос-кореляція золотих кодів $\approx \frac{1.4}{\sqrt{N}}$.

Послідовності Касамі генеруються шляхом децимації невеликого набору m-послідовностей. Нехай $s(n)$ m-послідовність довжини $N = 2^{2m} - 1$. Існує два типи послідовностей Касамі, мала множина та велика множина[9]. Для малої множини маємо

$$N = 2^n - 1, M = 2^{\frac{n}{2}} = \sqrt{N+1}, (\rho_{Kasamimax})^2 = (\frac{\sqrt{N+1}+1}{N})^2 \tag{2.17}$$

Наближення за допомогою ряду Пюізо дає

$$(\rho_{Kasamimax})^2 \approx \frac{1}{N} + 2(\frac{1}{N})^{\frac{3}{2}} + \frac{2}{N^2} + (\frac{1}{N})^{\frac{5}{2}} - \frac{1}{4}(\frac{1}{N})^{\frac{7}{2}} + \frac{1}{8}(\frac{1}{N})^{\frac{9}{2}} + O((\frac{1}{N})^{\frac{11}{2}}) \tag{2.18}$$

Використовуючи лише перший член ряду, маємо $(\rho_{Kasamimax})^2 \approx \frac{1}{N}$.

Для малої множини маємо

$$M = 2^n + 2^{\frac{n}{2}} = N + 1 + \sqrt{N+1}\ \rho_{GKasamimax} = (\frac{1+2^{\frac{n+2}{2}}}{N}),\ n = 2p \tag{2.19}$$

Та

$$M(\rho_{GKasamimax})^2 = (\frac{1+2\sqrt{N+1}}{N})^2 \approx \frac{4}{N} + 4(\frac{1}{N})^{\frac{3}{2}} + \frac{5}{N^2} + 2(\frac{1}{N})^{\frac{5}{2}} - \frac{1}{2}(\frac{1}{N})^{\frac{7}{2}} + \frac{1}{4}(\frac{1}{N})^{\frac{9}{2}} + O((\frac{1}{N})^{\frac{11}{2}}) \approx \frac{4}{N} \tag{2.20}$$

Таким чином, як коди Голда, так і послідовності Касамі є сім'ями послідовностей, що розповсюджуються, які мають низьку крос-кореляцію властивостями, що робить їх придатними для різних застосувань DS-SS. Їх максимальні значення кореляції наближаються до Велча для великих довжин послідовностей, що забезпечує міцну основу для їх широкого використання в системах зв'язку з широким спектром.

Також існують нові послідовності, що розповсюджуються, які існують як розширення добре відомих трирівневих сигналів. Ці нові послідовності генерують п'ятирівневу кореляційну функцію, що значно збільшує кардинальність набору послідовностей в N разів.

Нові коди утворюються шляхом комбінування трьох m-послідовностей, згенерованих LFSR, з певними переважними парами примітивних поліномів. Для $n = 2p + 1$ крос-кореляція приймає лише п'ять можливих значень[9].

$$\frac{-1-2^{\frac{n+2}{2}}+1}{2^n-1};\ \frac{-1-2^{\frac{n+1}{2}}}{2^n-1};\ \frac{-1}{2^n-1};\ \frac{-1-2^{\frac{n+1}{2}}}{2^n-1};\ \frac{-1+2^{\frac{n+1}{2}}+1}{2^n-1} \tag{2.21}$$

Максимальне абсолютне значення кореляційної функції для цих нових послідовностей є наступним

$$\rho_{Newmax} = \frac{1+2^{\frac{n+1}{2}+1}}{2^n-1},\ n = 2p + 1 \tag{2.22}$$

З параметрами

$$N = 2^n - 1,\ M = 2^{2n} + 2^n + 1 = (N+1)^2 + N + 2 \tag{2.23}$$

Таким чином, маємо

$$(\rho_{Newmax})^2 = (\frac{1+2^{\frac{n+1}{2}+1}}{2^n-1})^2 = (\frac{\sqrt{8(N+1)}+1}{N})^2 \tag{2.24}$$

Що при апроксимації рядом Пюізо дає результат

$$(\rho_{Newmax})^2 \approx \frac{8}{N} + 4\sqrt{2}(\frac{1}{N})^{\frac{3}{2}} + \frac{9}{N^2} + 2\sqrt{2}(\frac{1}{N})^{\frac{5}{2}} - \frac{(\frac{1}{N})^{\frac{7}{2}}}{\sqrt{2}} + \frac{(\frac{1}{N})^{\frac{9}{2}}}{2\sqrt{2}} + O((\frac{1}{N})^{\frac{11}{2}}) \quad (2.25)$$

Це вказує на те, що для великих значень N взаємна кореляція таких кодів становить приблизно $\frac{2.8}{\sqrt{N}}$ при апроксимації першим членом ряду.

Для глибокого аналізу нового класу послідовностей, що розповсюджуються, призначених для використання в SSIS, необхідно порівняти їх характеристики з існуючими кодами. В досягненнях у створенні таких послідовностей відзначається вирішальний етап: створення набору, який володіє п'ятизначною кореляційною функцією, що відповідає асимптотичній границі Велша. Це відповідність забезпечує оптимальний баланс між крос-кореляційними та автокореляційними властивостями, що є ключовим для мінімізації завад у спектрі зв'язку. Шляхом розкладання модуля кореляції та акцентування уваги на основному члені ряду, нові послідовності демонструють обмеження у $\frac{2.8}{\sqrt{N}}$. Це відображає подвоєння модуля кореляції порівняно з відомими кодами Голда. Незважаючи на це невелике збільшення, що робиться, відбувається значне збільшення кардинальності. Новий набір має кардинальність, що у N разів перевищує кардинальність великої множини кодів Касамі, що забезпечує суттєве збільшення кількості унікальних послідовностей, доступних для використання. Зведені оцінки кардинальності та модуля кореляції для розглянутих послідовностей наведено у таблиці 2.1[7].

Таблиця 2.1 - Порівняльний аналіз властивостей послідовностей розширення

	Коди Голда	Малий набір Касамі	Великий набір Касамі	Новий набір
Кардинальність, M	$N+2$	$\sqrt{N+1}$	$N+1$ $+\sqrt{N+1}$	$(N+2)^2+N$ $+2$
Максимальна кореляція(ρ_{max}, перший член)	$\frac{1.4}{\sqrt{N}}$	$\frac{1}{\sqrt{N}}$	$\frac{2}{\sqrt{N}}$	$\frac{2.8}{\sqrt{N}}$

Ця таблиця підкреслює значну перевагу, яку мають нові послідовності, що обіцяє революційно підвищити пропускну здатність асинхронних DS-SS та зменшити витрати на послуги зв'язку. Наприклад, використання нового набору може сприяти розширенню «м'якої ємності», що дозволить базовим станціям поступово нарощувати абонентську ємність абонентів при незначному зниженні якості обслуговування. Більше того, ці послідовності можуть значно підвищити пропускну здатність корисного навантаження в системах SSIS, тим самим розширюючи горизонти для безпечної і надійної передачі даних[7].

Новий набір послідовностей надає подвійну перевагу класам. По-перше, вони відкривають шлях до концепції «м'якої ємності», де базова станція може поступово збільшувати абонентську ємність з невеликими приростами, що сприяє динамічній адаптації до змінних потреб у зв'язку. По-друге, збагачена кардинальність послідовностей розповсюдження відкриває можливості для збільшення корисного навантаження в системі SSIS, що сприяє підвищенню швидкості вбудовування даних без ушкодження невидимості або цілісності вбудованого вмісту. Одним з ключових факторів у протистоянні геометричним атакам є характеристика некореляції цих послідовностей з їхніми зміщеними версіями. Після атаки система SSIS, обладнана цими послідовностями, може послідовно застосовувати геометричні перетворення до тих пір, поки не буде виявлено кореляційний пік. Цей пік сигналізує про правильне зворотне геометричне перетворення, що дозволяє відновити початкове вирівнювання стегоконтейнера і точно витягти приховане повідомлення. Цей стратегічний механізм захисту забезпечує, що навіть складні геометричні атаки не зможуть безповоротно пошкодити вбудовані дані. Замість цього, вони перетворюються у поле бою, де система SSIS, підкріплена новими послідовностями, може ефективно протистояти і нейтралізувати вплив атаки.

2.4 Генерація кодів Голда

Наразі відомо велике різноманіття цифрових послідовностей різних класів. При вивченні теорії їх побудови та властивостей необхідно мати на увазі завдання, які можна вирішити за допомогою обраних послідовностей. Серед різних застосувань цифрових послідовностей варто звернути увагу на наступні аспекти:

використання послідовностей у системах, де важливі автокореляційні властивості, використання груп послідовностей у системах, де важливі взаємнокореляційні властивості і використання послідовностей у системах з особливими вимогами до їх структури. У системах розширення спектра однією з ключових задач є виокремлення корисного сигналу з потоку даних, що надходять до приймача. Це означає, що вимагаються високі взаємнокореляційні властивості у використовуваних кодах. Рівень кореляції всіх послідовностей у кодовому ансамблі та кількість використовуваних послідовностей визначають максимальну кількість абонентів для даної системи.

Для генерації коду Голда використовується пара послідовностей максимальної довжини (m-послідовностей), які можуть бути отримані за допомогою регістра зсуву з лінійним зворотним зв'язком. Генерація m-послідовності довжини $N = 2\,m - 1$ здійснюється шляхом вибору примітивного полінома $h(x)$ ступеня m виду

$$h(x) = hOx^m + h1x^{m-1} + h2x^{m-2} + \cdots + hm - 1x + hm = \sum_{i=0}^{m} hix^{m-i} \quad (2.26)$$

Коефіцієнти h_i описують стани комірок пам'яті а у регістрі зсуву з лінійним зворотним зв'язком стан $h_i = 1$ відповідає підключенню виходу комірки пам'яті до ланцюжка суматорів, $h_i = 0$ відповідає його відсутності. Коефіцієнти першого й останнього осередку дорівнюють одиниці: $h_i = h_m = 1$.

Пошук примітивних поліномів є складною математичною проблемою. Задачу пошуку можна вирішити кількома методами, включаючи децимацію m-послідовності з коефіцієнтом децимації d, що призводить до отримання m-послідовності іншого типу, а також використання алгоритму Берлекемпа-Мессі для розв'язання системи лінійних рівнянь. Перевірка полінома включає дві основні операції: перевірку його непривідності та перевірку на примітивність.

Кількість примітивних поліномів ступеня m визначається наступним виразом[10].

$$Q = \frac{\phi(2^m - 1)}{m} \quad (2.27)$$

де $\phi(n)$ - функція Ейлера, що дорівнює кількості натуральних чисел, менших за n і взаємно простих із ним. Величина Q визначає кількість m-послідовностей заданої довжини $N = 2m - 1$. Число різних примітивних поліномів наведено у таблиці 2.2

Таблиця 2.2 - Число різних примітивних поліномів ступеня $m \leq 16$.

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Q	2	1	2	2	6	6	18	16	48	60	176	144	630	756	1800	2048

М-послідовності слугують базисом для формування інших важливих сімейств сигналів, одним із яких є коди Голда. Ансамбль послідовностей S може бути названий (l, σ) ансамблем, де l - довжина послідовностей S - обсяг ансамблю, а σ - максимальна кореляція між послідовностями всередині ансамблю. Ансамбль коду Голда є $(2^m - 1, 2^m + 1, 2^{(m+2)/2} + 1)$ ансамблем, а періодична взаємна кореляція між двома будь-якими послідовностями ансамблю має не більше трьох рівнів, тобто приймає тільки одне з трьох значень$(\pm 2, \frac{m+1}{2}, -1, -1)$. У склад ансамблю додаються як оригінальні копії m-послідовностей, так і результат побітового додавання їх за модулем 2, з подальшим циклічним зсувом однієї з послідовностей. Нехай L - оператор зсуву вліво, а m-послідовності a і b мають однакову довжину 7: $a = 11101000$ і $b = 1001011$. Тоді ансамбль коду Голда матиме вигляд[10]

Лістинг 2.1 – Утворення ансамблю кодів Голда

$$a = 11101000$$

$$b = 1001011$$

$$a \oplus b = 011111111$$

$$L(a) \oplus b = 01000010$$

$$L^2(a) \oplus b = 0011000$$

$$L^3(a) \oplus b = 1101100$$

$$L^4(a) \oplus b = 0000101$$

$$L^5(a) \oplus b = 1010110$$

$$L^6(a) \oplus b = 1110001$$

З кожним зростанням кількості примітивних поліномів ступеня m, збільшується і кількість можливих кодів Голда. Проте не для всіх комбінацій m-послідовностей однакової довжини можна сформувати ансамбль кодів Голда. Коди Голда утворюються з пари m-послідовностей, одна з яких отримана з іншої шляхом будь-якої з таких децимацій[10]

Лістинг 2.2 – Децимація m-послідовності

$$d = 2^k + 1, \gcd(k, m) = 1, k \leq \frac{m-1}{2}$$

$$d = 2^{2k} - 2^k + 1, \gcd(k, m) = 1, k \leq \frac{m-1}{2}, mod(3k, m) = 1$$

$$d = 2^{\frac{m-1}{2}} + 3$$

$$d = 2^{2k} + 2^k - 1, k = \frac{m-1}{4}, mod(m, 4) = 1$$

$$d = 2^{2k} + 2^{k-m} - 1, k = \frac{3m-1}{4}, mod(m, 4) = 3$$

Інверсія всіх коефіцієнтів децимації d для всіх випадків, перелічених вище.

$gcd(x, y)$ з наведених вище децимацій, являє собою функцію пошуку найменшого спільного дільника x і y.

$mod(x, y)$ - функція, що повертає число x за модулем y, тобто повертає залишок від ділення x на y.

Пошук відповідних пар включає такі етапи: спочатку вибирається довільний поліном порядку m. Потім за допомогою алгоритму Берлекемпа-Мессі з використанням усіх доступних децимацій обчислюється решта примітивних поліномів, які утворюють m-послідовності[10]. Наступним кроком є відбір дзеркальних копій отриманих пар, яким надається перевага. Для ступені полінома $m = 5$ всього існує $\frac{\phi(2^5-1)}{5} = 6$ різних примітивних поліномів

Лістинг 2.3 – Примітивні поліноми ступеня 5

$$x^5 + x^2 + 1, f_5 = (2)$$

$$x^5 + x^3 + 1, f_5 = (3)$$

$$x^5 + x^3 + x^2 + x + 1, f_5 = (3,2,1)$$

$$x^5 + x^4 + x^3 + x + 1, f_5 = (4,2,1)$$

$$x^5 + x^4 + x^3 + x + 1, f_5 = (4,3,1)$$

$$x^5 + x^4 + x^3 + x^2 + 1, f_5 = (4,3,2)$$

Поліноми можуть бути записані за ступенями без урахування старшого члена та одиниці. Для прикладу оберемо поліном $x^5 + x^3 + 1$ буде записаний як $f_5 = (3)$.

Унікальні індекси децимації з урахуванням інверсії кожної з них: d = 3,5,7,11. Шляхом децимації кожної з m-послідовностей, отримуємо ті самі послідовності в порядку. Отримані поліноми наведено у таблиці 2.3.

Таблиця 2.3 - Поліноми, отримані шляхом децимації вихідних m-послідовностей періоду $N = 2^5 - 1$

Початковий поліном	d	Отримані поліноми	d	Отримані поліноми
(2)	3	(4.3,2)	7	(3.2,1)
	5	(4,2,1)	1	(4.3,1)
(3)	3	(3,2,1)	7	(4,3,2)
	5	(4,3,1)	1	(4,2,1)
(3,2,1)	3	(4,3,1)	7	(4,2,1)
	5	(2)	1	(3)
(4,2,1)	3	(3)	7	(2)
	5	(3,2,1)	11	(4,3,2)
(4,3,1)	3	(2)	7	(3)
	5	(4,3,2)	11	(3,2,1)
(4,3,2)	3	(4,2,1)	7	(4,3,1)
	5	(3)	11	(2)

Відібравши дзеркальні копії знайдених пар, отримаємо наступні пари m послідовностей наведені у таблиці 2.4.

Таблиця 2.4 - Пари примітивних поліномів, що формують послідовності Голда довжини $N = 31$

№	f_m
1	(2), (3,2,1)
2	(2), (4,2,1)
3	(2), (4,3,1)
4	(2), (4,3,2)
5	(3), (3,2,1)

Продовження таблиці 2.4

6	(3), (4,2,1)
7	(3), (4,3,1)
8	(3), (4,3,2)
9	(3,2,1), (4,2,1)
10	(3,2,1), (4,3,1)
11	(4,2,1), (4,3,2)
12	(4,3,1), (4,3,2)

Зазвичай кількість пар, яким надають перевагу, є значно меншою, ніж загальна кількість всіх можливих пар m-послідовностей фіксованої довжини.

Для m, що діляться без залишку на 4, не існує кодів Голда. Однак існують подібні до них коди із числом піків періодичної взаємної кореляційної функції (ПВКФ) між послідовностями всередині ансамблю дорівнює 4. Так при $m = 8$ та $N = 255$ коди з 4 рівнями дорівнюють 16, а при $m = 12$ та $N = 4095$ кодів буде 144.

Наприклад, для довжини $N = 2^{10} - 1$ загальне число різних m-послідовностей становить 60, тоді як загальне число різних ансамблів кодів Голда складає $60^2 = 3600$. Проте лише 300 ансамблів відповідають умовам генерації та властивостям кодів Голда, маючи 3 рівні ПВКФ, що не перевищують значення $\sigma = 2^{(m+2)/2} + 1$ за модулем[10]. Таким чином число примітивних поліномів і ансамблів кодів Голда наведено у таблиці 2.5

Таблиця 2.5 - Число примітивних поліномів Q і ансамблів кодів Голда для фіксованої довжини N

m	N	Q	Коди Голда
5	31	6	12
6	63	6	6
7	127	18	90
8	255	16	-
9	511	48	288
10	1023	60	300

Продовження таблиці 2.5

11	2047	176	1936
12	4095	144	-
13	8191	630	8190
14	16383	756	6048
15	32767	1800	16200
16	65535	2048	-

Далі на таблиці 2.6 наведено пари примітивних поліномів.

Таблиця 2.6 - Пари примітивних поліномів, що формують послідовності Голда довжини $N = 1023$

№	f_m	№	f_m
1	(3),(6,5,3,2,1)	151	(8,5,4,3,2),(9,7,3)
2	(3),(7,3,1)	152	(8,5,4,3,2),(9,7,5,4,2)
3	(3),(7,6,5,2,1)	153	(8,5,4,3,2),(9,8,4,3,2)
4	(3),(8,3,2)	154	(8,5,4,3,2),(9,8,6,5,1)
5	(3),(8,5,1)	155	(8,6,1),(9,6,1)
6	(3),(8,7,6,5,2)	156	(8,6,1),(9,8,6,2,1)
7	(3),(9,6,5,4,3)	157	(8,6,1),(9,8,6,4,2)
8	(3),(9,8,6,3,2)	158	(8,6,1),(9,8,7,5,4)
9	(3),(9,8,6,4,2)	159	(8,6,1),(9,8,7,6,4,3,1)
10	(3),(9,8,7,6,4,3,1)	160	(8,6,4,2,1),(8,7,5)
11	(4,3,1),(6,5,3,2,1)	161	(8,6,4,2,1),(8,7,6,5,4,3,1)
12	(4,3,1),(7,6,5,4,3,2,1)	162	(8,6,4,2,1),(9,4,2)
13	(4,3,1),(8,7,2)	163	(8,6,4,2,1),(9,8,5)
14	(4,3,1), (8,7,6,5,4,2,1)	164	(8,6,4,2,1),(9,8,6,4,3)
15	(4,3,1),(9,5,2)	165	(8,6,4,2,1), (9,8,6,5,1)
16	(4,3,1), (9,6,5,4,3)	166	(8,6,5,3,1),(8,7,2)
17	(4,3,1),(9,7,6,4,1)	167	(8,6,5,3,1),(8,7,6,5,2)
18	(4,3,1),(9,7,6,4,3,2,1)	168	(8,6,5,3,1),(8,7,6,5,4,2,1)
19	(4,3,1), (9,7,6,5,4,3,2)	169	(8,6,5,3,1),(9,5,2)
20	(4,3,1), (9,8,5,4,3)	170	(8,6,5,3,1), (9,5,4,2,1)
21	(5,2,1),(7,3,1)	171	(8,6,5,3,1), (9,6,5,4,3)
22	(5,2,1),(7,6,2)	172	(8,6,5,3,1),(9,7,6,4,1)
23	(5,2,1),(8,6,1)	173	(8,6,5,3,1),(9,7,6,4,3,2,1)
24	(5,2,1),(9,5,4,2,1)	174	(8,6,5,3,1), (9,8,5,4,3)

Продовження таблиці 2.6

25	(5,2,1), (9,7,6,4,1)	175	(8,7,2),(8,7,4,2,1)
26	(5,2,1),(9,7,6,5,4,3,2)	176	(8,7,2),(9,5,2)
27	(5,2,1),(9,8,6,2,1)	177	(8,7,2),(9,7,3)
28	(5,2,1),(9,8,6,4,2)	178	(8,7,2),(9,8,4,3,2)
29	(5,2,1), (9,8,7,3,2)	179	(8,7,2),(9,8,5,4,3)
30	(5,2,1),(9,8,7,4,1)	180	(8,7,2),(9,8,7,6,5,4,1)
31	(5,3,2),(7,6,2)	181	(8,7,3,2,1),(8,7,6,5,4,3,1)
32	(5,3,2),(8,5,1)	182	(8,7,3,2,1), (9,6,1)
33	(5,3,2), (8,5,4,3,2)	183	(8,7,3,2,1),(9,6,5,4,3)
34	(5,3,2), (8,6,1)	184	(8,7,3,2,1), (9,6,5,4,3,2,1)
35	(5,3,2),(8,7,4,2,1)	185	(8,7,3,2,1),(9,8,5)
36	(5,3,2), (9,6,3,2,1)	186	(8,7,3,2,1), (9,8,6,5,1)
37	(5,3,2),(9,7,3)	187	(8,7,4,2,1),(9,7,6,4,3,2,1)
38	(5,3,2), (9,8,6,2,1)	188	(8,7,4,2,1), (9,8,4,3,2)
39	(5,3,2),(9,8,6,4,2)	189	(8,7,4,2,1),(9,8,6,5,4,3,2)
40	(5,3,2),(9,8,7,5,4)	190	(8,7,4,2,1), (9,8,7,5,4)
41	(6,5,2),(8,4,3)	191	(8,7,5),(8,7,6,5,2)
42	(6,5,2), (8,6,4,2,1)	192	(8,7,5),(9,4,2)
43	(6,5,2),(8,7,3,2,1)	193	(8,7,5),(9,5,2)
44	(6,5,2), (8,7,4,2,1)	194	(8,7,5),(9,8,4,2,1)
45	(6,5,2),(8,7,6,5,4,3,1)	195	(8,7,5),(9,8,6,3,2)
46	(6,5,2), (9,4,2)	196	(8,7,5),(9,8,7,4,1)
47	(6,5,2),(9,6,3,2,1)	197	(8,7,6,2,1),(8,7,6,5,2)
48	(6,5,2),(9,8,4,2,1)	198	(8,7,6,2,1),(9,8,6,2,1)
49	(6,5,2),(9,8,6,5,1)	199	(8,7,6,2,1),(9,8,6,3,2)
50	(6,5,2),(9,8,6,5,4,3,2)	200	(8,7,6,2,1),(9,8,7,6,4,3,1)
51	(6,5,3,2,1),(7,3,1)	201	(8,7,6,5,2),(9,4,2)
52	(6,5,3,2,1),(8,4,3)	202	(8,7,6,5,2),(9,5,4,2,1)
53	(6,5,3,2,1),(8,7,5)	203	(8,7,6,5,2),(9,8,6,3,2)
54	(6,5,3,2,1),(8,7,6,2,1)	204	(8,7,6,5,2),(9,8,6,4,3)
55	(6,5,3,2,1),(9,4,2)	205	(8,7,6,5,4,2,1),(9,5,2)
56	(6,5,3,2,1),(9,7,6,5,4,3,2)	206	(8,7,6,5,4,2,1),(9,6,1)
57	(6,5,3,2,1),(9,8,6,3,2)	207	(8,7,6,5,4,2,1),(9,6,5,4,3)
58	(6,5,3,2,1),(9,8,6,4,3)	208	(8,7,6,5,4,2,1),(9,6,5,4,3,2,1)
59	(7),(7,6,5,4,1)	209	(8,7,6,5,4,2,1),(9,8,5,4,3)
60	(7),(8,5,4,3,2)	210	(8,7,6,5,4,2,1),(9,8,6,3,2)

Продовження таблиці 2.6

61	(7),(8,6,4,2,1)	211	(8,7,6,5,4,3,1),(9,6,1)
62	(7),(8,7,2)	212	(8,7,6,5,4,3,1),(9,6,3,2,1)
63	(7),(8,7,4,2,1)	213	(8,7,6,5,4,3,1),(9,6,5,4,3,2,1)
64	(7),(9,5,2)	214	(8,7,6,5,4,3,1),(9,7,6)
65	(7),(9,7,3)	215	(8,7,6,5,4,3,1),(9,8,4,2,1)
66	(7),(9,7,6,4,3,2,1)	216	(8,7,6,5,4,3,1),(9,8,5)
67	(7),(9,8,5,4,3)	217	(8,7,6,5,4,3,1),(9,8,7,5,4)
68	(7),(9,8,7,5,4)	218	(9,4,1),(9,4,2)
69	(7,3,1),(8,3,2)	219	(9,4,1),(9,5,4,2,1)
70	(7,3,1),(8,7,5)	220	(9,4,1),(9,6,4,3,1)
71	(7,3,1),(8,7,6,2,1)	221	(9,4,1),(9,7,6,4,3,2,1)
72	(7,3,1),(8,7,6,5,2)	222	(9,4,1),(9,7,6,5,4,3,2)
73	(7,3,1),(9,7,6,4,1)	223	(9,4,1),(9,8,6,5,4,3,2)
74	(7,3,1),(9,8,6,4,3)	224	(9,4,1),(9,8,7,3,2)
75	(7,3,1),(9,8,7,6,4,3,1)	225	(9,4,1),(9,8,7,4,1)
76	(7,6,2),(7,6,4,2,1)	226	(9,4,1),(9,8,7,6,5,4,3)
77	(7,6,2),(8,3,2)	227	(9,4,2),(9,7,6,4,3,2,1)
78	(7,6,2),(8,5,4)	228	(9,4,2),(9,8,4,2,1)
79	(7,6,2),(8,5,4,3,2)	229	(9,4,2),(9,8,5)
80	(7,6,2),(9,6,5,4,3,2,1)	230	(9,4,2),(9,8,6,4,3)
81	(7,6,2),(9,8,6,2,1)	231	(9,5,2),(9,7,6,4,1)
82	(7,6,2),(9,8,6,4,2)	232	(9,5,2),(9,7,6,4,3,2,1)
83	(7,6,2),(9,8,7,5,4)	233	(9,5,2),(9,8,4,3,2)
84	(7,6,4,2,1),(7,6,5,2,1)	234	(9,5,2),(9,8,7,4,1)
85	(7,6,4,2,1),(8,5,4,3,2)	235	(9,5,4,2,1),(9,8,6,2,1)
86	(7,6,4,2,1),(8,6,1)	236	(9,5,4,2,1),(9,8,6,4,2)
87	(7,6,4,2,1),(8,7,3,2,1)	237	(9,5,4,2,1),(9,8,7,3,2)
88	(7,6,4,2,1),(8,7,4,2,1)	238	(9,5,4,2,1),(9,8,7,4,1)
89	(7,6,4,2,1),(9,7,3)	239	(9,5,4,2,1),(9,8,7,6,5,4,1)
90	(7,6,4,2,1),(9,8,6,2,1)	240	(9,6,1),(9,6,3,2,1)
91	(7,6,4,2,1),(9,8,6,4,2)	241	(9,6,1),(9,6,5,4,3)
92	(7,6,4,2,1),(9,8,7,5,4)	242	(9,6,1),(9,7,6,4,1)
93	(7,6,5,2,1),(8,3,2)	243	(9,6,1),(9,8,6,5,1)
94	(7,6,5,2,1),(8,7,3,2,1)	244	(9,6,1),(9,8,7,6,4,3,1)
95	(7,6,5,2,1),(8,7,6,2,1)	245	(9,6,3,2,1),(9,6,5,4,3)
96	(7,6,5,2,1),(9,6,4,3,1)	246	(9,6,3,2,1),(9,6,5,4,3,2,1)

Продовження таблиці 2.6

97	(7,6,5,2,1),(9,7,5,4,2)	247	(9,6,3,2,1),(9,8,5)
98	(7,6,5,2,1),(9,7,6)	248	(9,6,3,2,1),(9,8,6,5,1)
99	(7,6,5,2,1),(9,8,6,5,4,3,2)	249	(9,6,4,3,1),(9,7,3)
100	(7,6,5,2,1),(9,8,7,6,4,3,1)	250	(9,6,4,3,1),(9,7,5,4,2)
101	(7,6,5,4,1),(8,6,4,2,1)	251	(9,6,4,3,1),(9,7,6)
102	(7,6,5,4,1),(9,4,1)	252	(9,6,4,3,1),(9,8,5)
103	(7,6,5,4,1),(9,6,4,3,1)	253	(9,6,4,3,1),(9,8,7,6,5,4,1)
104	(7,6,5,4,1),(9,7,5,4,2)	254	(9,6,4,3,1),(9,8,7,6,5,4,3)
105	(7,6,5,4,1),(9,7,6)	255	(9,6,5,4,3),(9,6,5,4,3,2,1)
106	(7,6,5,4,1),(9,8,6,5,4,3,2)	256	(9,6,5,4,3),(9,7,6,4,1)
107	(7,6,5,4,1),(9,8,7,3,2)	257	(9,6,5,4,3),(9,8,6,4,2)
108	(7,6,5,4,1),(9,8,7,4,1)	258	(9,6,5,4,3,2,1),(9,7,6,4,1)
109	(7,6,5,4,1),(9,8,7,6,5,4,1)	259	(9,6,5,4,3,2,1),(9,8,6,5,1)
110	(7,6,5,4,3,2,1),(8,6,5,3,1)	260	(9,7,3),(9,7,6,4,3,2,1)
111	(7,6,5,4,3,2,1),(8,7,3,2,1)	261	(9,7,3),(9,8,4,3,2)
112	(7,6,5,4,3,2,1),(8,7,6,2,1)	262	(9,7,3),(9,8,5)
113	(7,6,5,4,3,2,1),(8,7,6,5,4,2,1)	263	(9,7,3),(9,8,7,5,4)
114	(7,6,5,4,3,2,1),(9,6,1)	264	(9,7,5,4,2),(9,8,6,5,1)
115	(7,6,5,4,3,2,1),(9,6,3,2,1)	265	(9,7,5,4,2),(9,8,6,5,4,3,2)
116	(7,6,5,4,3,2,1),(9,6,5,4,3,2,1)	266	(9,7,5,4,2),(9,8,7,6,4,3,1)
117	(7,6,5,4,3,2,1),(9,7,6,4,1)	267	(9,7,5,4,2),(9,8,7,6,5,4,3)
118	(7,6,5,4,3,2,1),(9,8,6,2,1)	268	(9,7,6),(9,8,6,5,4,3,2)
119	(8,3,2),(8,5,1)	269	(9,7,6),(9,8,7,5,4)
120	(8,3,2),(8,7,6,2,1)	270	(9,7,6),(9,8,7,6,4,3,1)
121	(8,3,2),(9,6,5,4,3,2,1)	271	(9,7,6),(9,8,7,6,5,4,3)
122	(8,3,2),(9,7,5,4,2)	272	(9,7,6,4,1),(9,8,5,4,3)
123	(8,3,2),(9,7,6)	273	(9,7,6,4,3,2,1),(9,8,4,3,2)
124	(8,3,2),(9,8,6,3,2)	274	(9,7,6,4,3,2,1),(9,8,5,4,3)
125	(8,4,3),(8,6,4,2,1)	275	(9,7,6,5,4,3,2),(9,8,6,2,1)
126	(8,4,3),(8,7,2)	276	(9,7,6,5,4,3,2),(9,8,6,4,2)
127	(8, 4, 3), (8, 7, 5)	277	(9, 7, 6, 5, 4, 3, 2), (9, 8, 7, 3, 2)
128	(8, 4, 3), (8, 7, 6, 5, 2)	278	(9, 7, 6, 5, 4, 3, 2), (9, 8, 7, 4, 1)
129	(8, 4, 3), (9, 8, 4, 2, 1)	279	(9, 7, 6, 5, 4, 3, 2), (9, 8, 7, 6, 5, 4, 1)
130	(8, 4, 3), (9, 8, 5)	280	(9, 8, 4, 2, 1), (9, 8, 4, 3, 2)
131	(8, 4, 3), (9, 8, 6, 4, 3)	281	(9, 8, 4, 2, 1), (9, 8, 5)

Продовження таблиці 2.6

113	(7,6,5,4,3,2,1),(8,7,6,5,4,2,1)	263	(9,7,3),(9,8,7,5,4)
114	(7,6,5,4,3,2,1),(9,6,1)	264	(9,7,5,4,2),(9,8,6,5,1)
115	(7,6,5,4,3,2,1),(9,6,3,2,1)	265	(9,7,5,4,2),(9,8,6,5,4,3,2)
116	(7,6,5,4,3,2,1),(9,6,5,4,3,2,1)	266	(9,7,5,4,2),(9,8,7,6,4,3,1)
117	(7,6,5,4,3,2,1),(9,7,6,4,1)	267	(9,7,5,4,2),(9,8,7,6,5,4,3)
118	(7,6,5,4,3,2,1),(9,8,6,2,1)	268	(9,7,6),(9,8,6,5,4,3,2)
119	(8,3,2),(8,5,1)	269	(9,7,6),(9,8,7,5,4)
120	(8,3,2),(8,7,6,2,1)	270	(9,7,6),(9,8,7,6,4,3,1)
121	(8,3,2),(9,6,5,4,3,2,1)	271	(9,7,6),(9,8,7,6,5,4,3)
122	(8,3,2),(9,7,5,4,2)	272	(9,7,6,4,1),(9,8,5,4,3)
123	(8,3,2),(9,7,6)	273	(9,7,6,4,3,2,1),(9,8,4,3,2)
124	(8,3,2),(9,8,6,3,2)	274	(9,7,6,4,3,2,1),(9,8,5,4,3)
125	(8,4,3),(8,6,4,2,1)	275	(9,7,6,5,4,3,2),(9,8,6,2,1)
126	(8,4,3),(8,7,2)	276	(9,7,6,5,4,3,2),(9,8,6,4,2)
127	(8, 4, 3), (8, 7, 5)	277	(9, 7, 6, 5, 4, 3, 2), (9, 8, 7, 3, 2)
128	(8, 4, 3), (8, 7, 6, 5, 2)	278	(9, 7, 6, 5, 4, 3, 2), (9, 8, 7, 4, 1)
129	(8, 4, 3), (9, 8, 4, 2, 1)	279	(9, 7, 6, 5, 4, 3, 2), (9, 8, 7, 6, 5, 4, 1)
130	(8, 4, 3), (9, 8, 5)	280	(9, 8, 4, 2, 1), (9, 8, 4, 3, 2)
131	(8, 4, 3), (9, 8, 6, 4, 3)	281	(9, 8, 4, 2, 1), (9, 8, 5)
132	(8, 4, 3), (9, 8, 7, 6, 5, 4, 1)	282	(9, 8, 4, 2, 1), (9, 8, 6, 4, 3)
133	(8,4,3),(9,8,6,5,4,1)	283	(9,8,4,2,1),(9,8,6,5,1)
134	(8,4,3), (9,8,7,3,2)	284	(9,8,4,2,1),(9,8,6,5,4,3,2)
135	(8,4,3), (9,8,7,4,1)	285	(9,8,4,2,1), (9,8,7,6,4,3,1)
136	(8,5,1), (8,5,4,3,2)	286	(9,8,5), (9,8,6,4,3)
137	(8,5,1), (8,7,3,2,1)	287	(9,8,5), (9,8,6,5,1)
138	(8,5,1), (8,7,6,5,4,2,1)	288	(9,8,5), (9,8,7,6,4,3,1)
139	(8,5,1), (9,6,5,4,3,2,1)	289	(9,8,6,2,1),(9,8,6,4,2)
140	(8,5,1), (9,7,6,5,4,3,2)	290	(9,8,6,2,1),(9,8,7,4,1)
141	(8,5,4,3,2), (8,6,1)	291	(9,8,6,3,2),(9,8,6,5,1)
142	(8,5,4,3,2), (8,7,4,2,1)	292	(9,8,6,3,2),(9,8,7,6,4,3,1)
143	(8,5,4,3,2), (9,5,2)	293	(9,8,6,4,2),(9,8,7,3,2)
144	(8,5,4,3,2), (9,6,1)	294	(9,8,6,4,2),(9,8,7,4,1)
145	(8,5,4,3,2), (9,7,3)	295	(9,8,6,5,1),(9,8,7,5,4)
146	(8,5,4,3,2), (9,7,5,4,2)	296	(9,8,6,5,1),(9,8,7,6,4,3,1)
147	(8,5,4,3,2), (9,8,6,5,1)	297	(9,8,7,3,2),(9,8,7,6,4,3,1)
148	(8,6,1), (8,6,4,2,1)	298	(9,8,7,4,1),(9,8,7,6,5,4,3)

Продовження таблиці 2.6

149	(8,6,1), (9,6,3,2,1)	299	(9,8,7,5,4), (9,8,7,6,4,3,1)
150	(8,6,1), (9,8,6,2,1)	300	(9,8,7,5,4,1), (9,8,7,6,5,4,1)

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ МЕТОДИКИ АДАПТИВНОГО ВБУДОВУВАННЯ ЦИФРОВИХ ВОДЯНИХ ЗНАКІВ В АУДІОФАЙЛИ

У традиційному зв'язку інформація зазвичай концентрується у вузькому діапазоні частот для економії смуги пропускання і потужності. Однак методи розширення спектру спрямовані на захист інформації шляхом поширення закодованих даних по всьому частотному спектру, що робить їх більш стійкими до перешкод.

3.1 Огляд сутності досліджуваної методики адаптивного вбудовування цифрових водяних знаків в аудіофайли

Один з таких методів, розширення спектру сигналу прямої послідовності (DS-SS), передбачає розширення сигналу даних шляхом множення його на послідовність максимальної довжини, модульовану з відомою частотою. При застосуванні до дискретних аудіосигналів частота дискретизації може використовуватися як модулююча частота, що спрощує фазову синхронізацію і дозволяє підвищити швидкість передачі даних. Розширення спектру сигналу випадкової послідовності - це ще один метод розширення спектру, який вимагає використання одного і того ж псевдовипадкового ключа для шифрування і дешифрування. В ідеалі, цей ключ має пласку частотну характеристику у всьому діапазоні частот, подібно до білого шуму. У контексті аудіосигналів, даний метод передбачає множення сигналу даних як на несучий сигнал, так і на послідовність псевдовипадкового шуму. Це розширює спектр даних по всій доступній смузі пропускання. Отримана розширена послідовність даних потім послаблюється і додається до вихідного сигналу як випадковий шум. Метод розширення спектру сигналу прямою послідовністю заснований на двійковій фазовій маніпуляції, через те що фаза сигналу псевдовипадкової послідовності постійно по черзі змінюється з фазою модульованої двійкової послідовності.

Під час процесу екстракції значення фази ϕ_0 $\phi_0 + \pi$ інтерпретуються як двійкові цифри «0» і «1» відповідно, які були використані для кодування вихідних даних. Цей метод ґрунтується на наступних умовах[11].

Псевдовипадковий ключ є М-послідовністю, яка забезпечує відносно плоский частотний спектр завдяки своїй максимальній довжині та рівномірному розподілу комбінацій. При цьому одержувач знає ключ шифрування, досягнув синхронізації сигналів і знає початкову та кінцеву точки розширених даних, знає частоту елементарних повідомлень, швидкість передачі даних і частоту несучого сигналу.

На відміну від фазового кодування, метод з прямим розширенням спектру вводить в аудіосигнал адитивний випадковий шум. Щоб рівень шуму був низьким і нечутним, розширений код послаблюється приблизно до 0,5% від динамічного діапазону аудіофайлу контейнера[11]. Цілісність двійкової послідовності забезпечується поєднанням простого повторення та методів кодування з корекцією помилок. Короткі сегменти двійкового коду об'єднуються і переплітаються з сигналом аудіоконтейнера для мінімізації перехідних шумів. Під час декодування весь сегмент усереднюється для досягнення цієї мети.

Теоретичною основою завадозахищеного зв’язку є відома теорема Шеннона про пропускну здатність каналу зв’язку. Вона стверджує, що за умови, що швидкість передачі інформації R менша за пропускну здатність каналу зв’язку C, можна знайти такі методи кодування, які забезпечать передачу інформації з потрібною якістю при будь-якому, навіть дуже малому співвідношенні потужності сигналу $\mathrm{P_C}$ до потужності завади $\mathrm{P_П}$. Хоча теорема не визначає конкретні методи кодування, вона чітко окреслює шлях до досягнення необхідної якості передачі.

За теоремою Шеннона пропускна здатність каналу зв’язку буде дорівнювати[11].

$$C = \Delta F_k log_2(1 + \frac{\mathrm{P_C}}{\mathrm{P_П}}) \tag{3.1}$$

Де ΔF_k – ширина смуги пропускання каналу.

Якщо поділити обидві частини попереднього рівняння на ширину смуги пропускання каналу, тобто ΔF_k, і при цьому змінивши основу логарифма то отримаємо наступну рівність[11].

$$\frac{C}{\Delta F_k} = 1.44 * ln(1 + \frac{\mathrm{P_C}}{\mathrm{P_П}}) \tag{3.2}$$

Якщо припустити, що $(1+\frac{\text{Р}_\text{С}}{\text{Р}_\text{П}})$ буде менше за одиницю, то в такому випадку це буде корисним для захищених каналів радіозв'язку, то тоді рівність матиме наступний вигляд[11].

$$\frac{C}{\Delta F_k} = 1.44 * \left[\frac{\text{Р}_\text{С}}{\text{Р}_\text{П}} - \frac{1}{2}\left(\frac{\text{Р}_\text{С}}{\text{Р}_\text{П}}\right)^2 + \frac{1}{3}\left(\frac{\text{Р}_\text{С}}{\text{Р}_\text{П}}\right)^3 k\right] \quad (3.3)$$

При скороченні та враховуючи вище припущене твердження, можна записати рівність наступним чином[11].

$$\frac{C}{\Delta F_k} = 1.44 * \frac{\text{Р}_\text{С}}{\text{Р}_\text{П}} \quad (3.4)$$

Отже в такому випадку задається можливість досягнення необхідної якості передачі сигналу, враховуючи те що $\frac{\text{Р}_\text{С}}{\text{Р}_\text{П}}$ може набувати наскільки завгодно мінімальне значення, а якщо вважати, що ширина смуги пропускання каналу буде дорівнювати ширині спектра використовуваного сигналу, то зі зменшенням потужності сигналу до потужності перешкоди, виявляється що необхідно використовувати методи розширення спектра сигналу.

Структурно можна зобразити модель системи в якій відбувається прийом та передача інформації наступним чином, що зображено на рисунку 3.1[11].

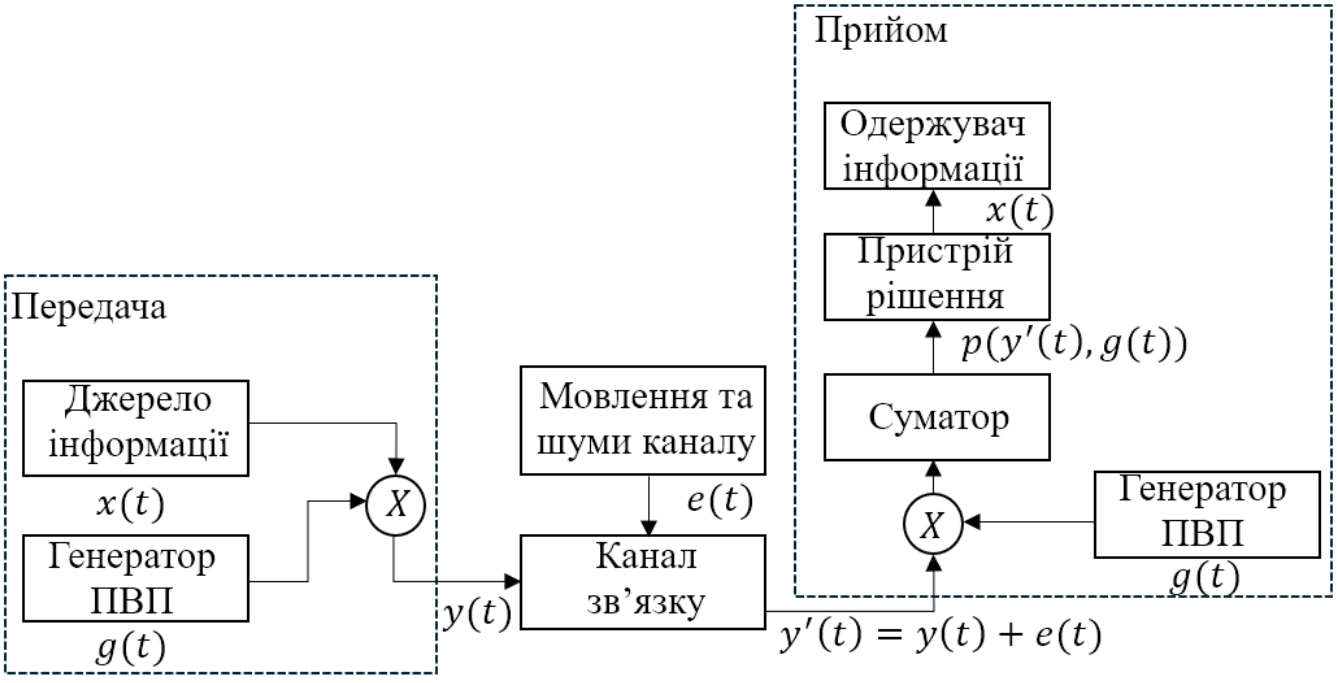

Рисунок 3.1 – Структурна схема передачі та прийому інформації з використанням прямого розширення спектру

Для того, щоб реалізувати методику адаптивного вбудовування цифрового водяного знаку в аудіофайл було принято рішення використовувати мову програмування Java. Основа методики полягає у використанні розглянутих раніше кодів Голда. Основними файлами, що використовуються програмою є аудіофайл, що виступає у ролі контейнера і текстовий файл, в якому міститься текст, що буде вважатись за цифровий водяний знак(ЦВЗ). Суть практичної реалізації полягає у тому, щоб прочитати аудіофайл та текстовий файл у вигляді двійкових послідовностей, потім згенерувати послідовність Голда та за допомогою неї вписати двійкову послідовність ЦВЗ до аудіофайлу, після чого виконати його збереження. Після необхідно провести аналіз отриманого аудіофайлу та початкового.

3.2 Практична реалізація досліджуваної методики

Реалізація генерації послідовності кодів Голда полягає у створенні m-послідовностей з подальшим їх об’єднанням між собою певним чином в одну послідовність, яка і буде послідовністю кодів Голда. М-послідовність має приблизно рівну кількість нулів і одиниць, а також хорошу автокореляцію. На початку створюються дві m-послідовностей із даної максимальної довжини та поліномів. Генерація даних послідовностей відбувається за допомогою регістру лінійного здвигу, або LFSR, суть якого полягає у тому, що кожен біт на кожному такті зсувається на одну позицію вправо. Новий біт, що входить зліва, обчислюється як виключне АБО певних бітів регістра. Ці біти, що беруть участь в операції XOR, називаються відводами і визначаються многочленом зворотного зв’язку. Многочлен зворотного зв’язку являє собою бітову маску, де кожен біт відповідає відведенню від регістра зсуву. Наприклад, многочлен $x^5 + x^3 + x^2 + x + 1$ (представлений у коді як $0x25$) означає, що відводи беруть із 5-го, 3-го, 2-го і 1-го бітів регістра. Регістр зсуву ініціалізується деяким початковим станом, а саме всі біти дорівнюють 0, тоді як перший біт дорівнює 1. Обчислення зворотного зв’язку відбувається за допомогою зчитування значення попереднього

біта(feedback), після чого обчислюється операція XOR між feedback і бітами з відводів, визначених многочленом. Після чого весь регістр зсувається вправо, а обчислене значення feedback записується в найлівіший біт регістра. Для генерації кодів Голда використовуються дві m-послідовності, отримані за допомогою LFSR з різними многочленами зворотного зв'язку. Ці многочлени називаються кращою парою, оскільки вони володіють певними математичними властивостями, що забезпечують низьку взаємну кореляцію між кодами Голда, що генеруються. Алгоритм отримання значення з послідовності Голда полягає у наступному. На початку з раніше згенерованої першої m-послідовності за індексом i обирається біт і з другої послідовності обирається біт зміщений на 10 пунктів відносно загального індексу. Потім результат береться за модулем довжини m-послідовності. Це забезпечує циклічний зсув індексу, тобто якщо $i + 10$ перевищує довжину послідовності, індекс повернеться до початку. Далі між значеннями двох послідовностей, застосовується операція XOR, після чого встановлюється i біт у послідовності кодів Голда.

Далі виконується операція вбудовування ЦВЗ в аудіофайл з використанням згенерованих кодів Голда. Суть полягає в тому, щоб в двійкову послідовність записати іншу двійкову послідовність, але використовуючи певний інтервал між символами та послідовність кодів Голда. У створеній реалізації один біт ЦВЗ встановлюється через кожні 100 біт аудіофайлу. За циклом перевіряються біти ЦВЗ, у випадку якщо біт дорівнює одиниці, то обирається відповідне за індексом значення біта з послідовності кодів Голда і відповідно біт з аудіоданих, замінюється на відповідний біт із послідовності Голда. У випадку якщо біт ЦВЗ є нулем, то в такому випадку відповідний за індексом біт в аудіофайлі також встановлюється в нульове значення. Після запису всіх даних, відбувається збереження нового аудіофайлу під назвою «watermarked.wav» до директорії створеного проекту.

3.3 Аналіз отриманих даних з практичної реалізації досліджуваної методики

Після успішного вбудовування ЦВЗ в аудіофайл проводиться його перевірка за допомогою декількох етапів. Першим доцільним етапом буде прослуховування

початкового аудіофайлу та отриманого після вбудовування. Як і очікувалось, жодних змін у якості, гучності, розмірі файлів немає жодної різниці, отже вбудований цифровий водяний знак не є помітним для людини, яка не використовує детальний аналіз файлу, що власне і відповідає основній меті поставленої задачі з приховування даних.

Далі виконується видобування ЦВЗ із аудіофайлу за допомогою зворотнього алгоритму. В даному випадку використовується зчитування аудіофайлу, перетворення його на послідовність бітів та за допомогою використання раніше утвореної послідовності Голда, відбувається покрове видобування даних ЦВЗ, таким же чином як і при запису. Результатом є вивід на екран початкового тексту ЦВЗ та результат його видобування, як видно даний етап аналізу проходить успішно, адже текст із аудіофайлу із ЦВЗ збігається з початковим цифровим знаком. Результат виконання даного етапу перевірки наведено на рисунку 3.2.

```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
1) За допомогою зворотнього алгоритму вбудовування, видобувається ЦВЗ з аудіофайлу
Оригінальний текст ЦВЗ: Lorem ipsum dolor sit amet
Видобутий водяний цифровий знак: Lorem ipsum dolor sit amet
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Рисунок 3.2 – Результат видобування прихованого ЦВЗ

Після доречною перевіркою є певна атака методом «грубої сили». Тобто обираються два файли і побітово перевіряються всі їх біти. Так як виявити вбудований ЦВЗ не вдається візуально чи аудіально, то можна виводити повідомлення про те коли біти не збігаються між собою. Якщо у методі використовувати аудіофайл із ЦВЗ та без нього, то отримуємо повідомлення, що біти у файлах відрізняються, якщо ж взяти до перевірки два однакових файла, то в такому випадку дійсно різниці між бітами не буде. Результат виконання даного етапу перевірки наведено на рисунку 3.3.

```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
2) Аналіз даних шляхом ітеративного перебору всіх бітів з двох аудіофайлів,
нижче виведено саме ті біти які відрізняються між двома аудіофайлами,
 що доводить успішне вбудування ЦВЗ
РЕЗУЛЬТАТ ПОБІТОВОЇ ПЕРЕВІРКИ: Виявлено різницю в бітах файлів!

2.1) Додаткова перевірка шляхом порівняння між собою двох однакових файлів за допомогою того ж методу
Порівняння двох однакових аудіофайлів без ЦВЗ
РЕЗУЛЬТАТ ПОБІТОВОЇ ПЕРЕВІРКИ: Файли однакові, різниці між бітами файлів немає!

Порівняння двох однакових аудіофайлів із ЦВЗ
РЕЗУЛЬТАТ ПОБІТОВОЇ ПЕРЕВІРКИ: Файли однакові, різниці між бітами файлів немає!
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Рисунок 3.3 – Результат видобування прихованого ЦВЗ

Далі проводиться кореляційний аналіз, що дає змогу визначити ступінь взаємозв’язку між двома наборами даних. Під час вбудовування ЦВЗ в аудіофайл, вихідні дані змінюються певним чином. Після вилучення передбачуваного ЦВЗ з аудіофайлу ми отримуємо два набори даних, саме оригінальні дані аудіофайлу та дані, що мають містити ЦВЗ. Якщо ЦВЗ був успішно вбудований, то витягнуті дані мають бути схожі на оригінальні дані. Цю схожість можна виміряти за допомогою коефіцієнта кореляції. Що вища кореляція, то більша ймовірність, що витягнуті дані дійсно містять ЦВЗ. В даній реалізації, кореляційних аналіз побудований на основі коефіцієнту кореляції Пірсона, що є статистичною мірою, який показує ступінь лінійної залежності між двома змінними. Даний коефіцієнт набуває значень від -1 до 1. Значення 1 вказує на повну позитивну кореляція. Це означає, що при збільшенні однієї змінної інша також збільшується пропорційно. При значенні 0 вказується на відсутність кореляції. Змінні незалежні одна від одної, і зміна однієї не впливає на іншу. Повна негативна кореляція відбувається при значенні -1. Тобто при збільшенні однієї змінної інша зменшується пропорційно. Коефіцієнт кореляції Пірсона вимірює тільки лінійну залежність. Якщо залежність нелінійна, коефіцієнт може бути близький до нуля, навіть якщо дані пов’язані між собою. Коефіцієнт кореляції не має на меті однозначно встановити причинно-наслідковий зв’язок між даними, а лише те, наскільки сильно дані змінюються разом. За результатами аналізу було отримане високе значення коефіцієнта кореляції, тобто близьке до 1. Це вказує на те, що витягнутий водяний знак дуже схожий на вихідний, що

підтверджує успішність вбудовування і вилучення. Результат виконання даного етапу перевірки наведено на рисунку 3.4.

```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
3) Проведення кореляційного аналізу між двома наборами даних
(аудіофайл з ЦВЗ та без нього) з допустимим порогом 0.8
Значення кореляції в допустимому діапазоні: 0.990686576282994).
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Рисунок 3.4 – Результат кореляційного аналізу

Далі проводиться аналіз статистичних характеристик аудіофайлів. Серед них розглянуто п’ять основних параметрів.

Середнє значення амплітуд усіх семплів в аудіофайлі. Відображає загальну «гучність» або «енергію» сигналу. Що вище середнє значення, то голосніший звук. На цей параметр впливають гучність запису, баланс інструментів у міксі, наявність постійної складової шуму. Збільшення середнього значення на 0.166 може свідчити про те, що в процесі вбудовування ЦВЗ в аудіосигнал було внесено невеликі позитивні значення. Це пов’язано з особливостями алгоритму вбудовування, який змінює амплітуди деяких семплів.

Дисперсія показує наскільки сильно розкидані значення амплітуд відносно середнього значення. Велика дисперсія означає ширший діапазон гучності звуку, від тихих до гучних ділянок. Мала дисперсія свідчить про більш рівномірне звучання. На цей параметр впливають динамічний діапазон запису, наявність різких перепадів гучності, тип музичного жанру. Незначне збільшення дисперсії говорить про те, що розкид значень амплітуд навколо середнього трохи збільшився після вбудовування ЦВЗ. Це наслідок додавання невеликого шуму або випадкових змін до сигналу, власне випадкові зміни і є результатом роботи алгоритму з вбудовування ЦВЗ.

Стандартне відхилення, являє собою корінь квадратний із дисперсії. Більш зручна для інтерпретації міра розкиду амплітуд. Аналогічно дисперсії, велике стандартне відхилення вказує на широкий діапазон гучності. Збільшення

стандартного відхилення на 0.031 також пов’язане зі збільшенням дисперсії і підтверджує, що розкид значень амплітуд став трохи більшим.

Параметр «асиметрія» показує симетричність розподілу амплітуд щодо середнього значення. За позитивної асиметрія, розподіл зміщений вправо, тобто більше гучних звуків, ніж тихих. За негативної асиметрії розподіл зміщений вліво, тобто більше тихих звуків, ніж гучних. На цей параметр впливає тип аудіосигналу, тобто музика, мова, шум, наявність спотворень.

Ексцес показує «гостроту піку», або гучні звуки та «важкість хвостів», або тихі звуки та їхній розподіл амплітуд. Високий ексцес вказує на одиничний «гострий пік» і «довгі хвости», тобто багато тихих звуків і кілька дуже гучних. За низького ексцесу спостерігається плоска вершина і «короткі хвости», тобто амплітуди більш рівномірно розподілені. На цей параметр впливає тип аудіосигналу та наявність компресії, тобто зменшення динамічного діапазону. Незначні зміни останніх двох параметрів параметрів вказують на те, що вбудовування ЦВЗ практично не вплинуло на форму розподілу амплітуд. Це означає, що водяний знак не вніс істотних спотворень в аудіосигнал. Результат виконання даного етапу перевірки наведено у таблиці 3.1.

Таблиця 3.1 – Порівняння статистичних даних двох аудіофайлів

Параметр	Оригінальний файл	Файл із ЦВЗ	Різниця
Середнє значення амплітуд	-6.043	-5.877	+0.166
Дисперсія	70693000	70694000	+1000
Стандартне відхилення	8407.93	8407.96	+0.031
Асиметрія	0.2558	0.2557	-0.0001
Ексцес	1.0771	1.0770	-0.0001

Кінцевим етапом аналізу отриманих даних з є побудова графіків та осцилограм. Для цього використовується бібліотека «jFreeChart». Дані отримані з аудіофайлів заносяться до графіків, що автоматично будуються за допомогою бібліотеки та зберігаються у вигляді зображень. Поступово будується графік

Осцилограма являє собою графік залежності амплітуди звукового сигналу відносно часу та дає змогу візуально оцінити форму звукової хвилі та її зміни в часі.

За осцилограмою можна визначити гучність звуку, оскільки чим більша амплітуда хвилі, тим голосніший звук. Частоту звуку, чим частіше хвиля коливається, тим вища частота звуку. Наявність пауз і тиші, ділянки з нульовою амплітудою відповідають тиші. Характер звуку, тобто за формою хвилі можна визначити, чи є звук музичним, мовним, шумом тощо.

У контексті вбудовування ЦВЗ, порівнюючи осцилограми вихідного і модифікованого аудіофайлів, можна візуально оцінити, наскільки сильно вбудовування ЦВЗ вплинуло на звуковий сигнал. В ідеальному випадку, відмінності мають бути мінімальними і непомітними для ока. У нашому випадку при візуальному порівнянні двох осцилограм зміни помітні лише на самому початку, що саме і може свідчити про наявність ЦВЗ в аудіофайлі. Результат виконання даного етапу перевірки наведено на рисунку 3.5 та на рисунку 3.6.

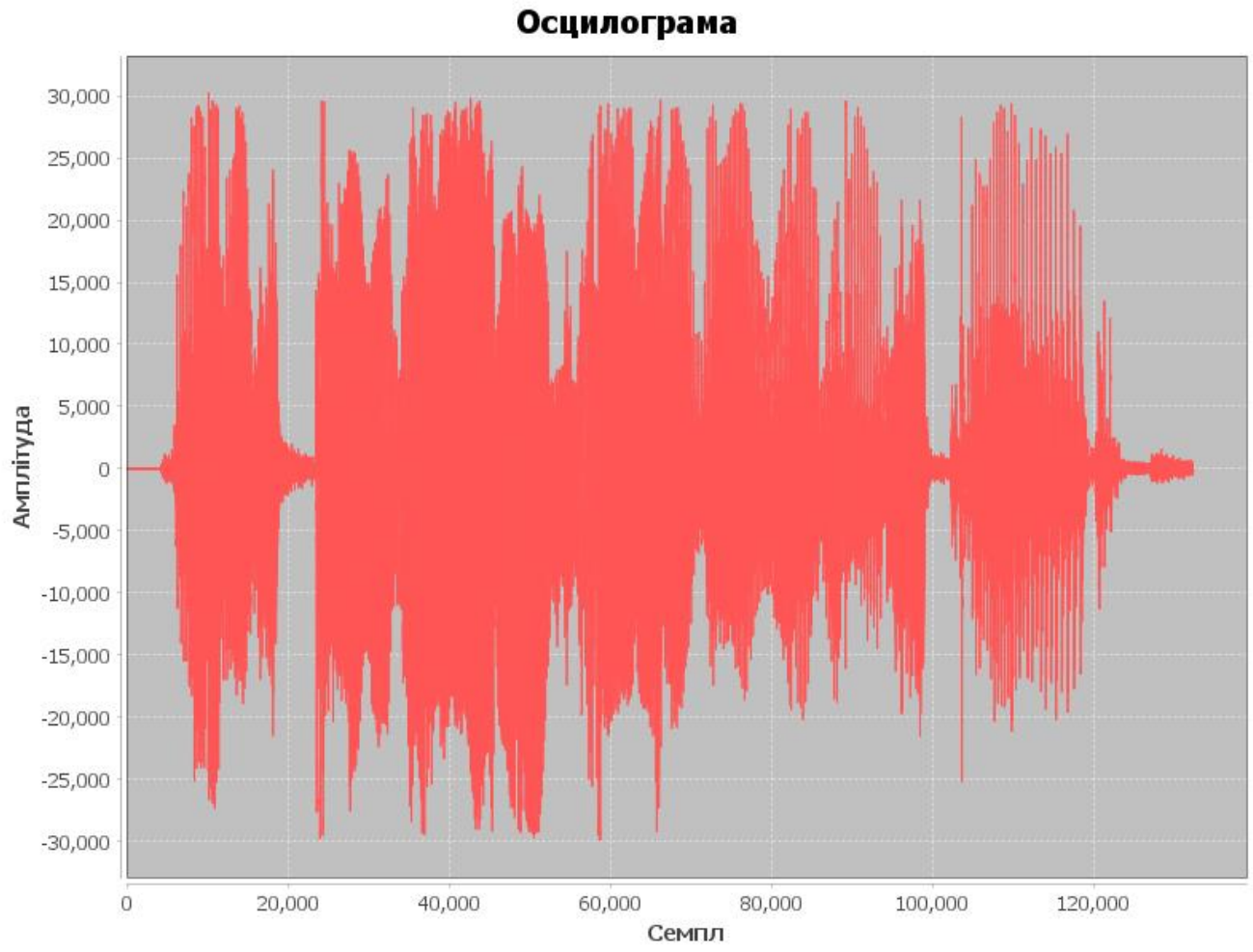

Рисунок 3.5 – Осцилограма оригінального аудіофайлу

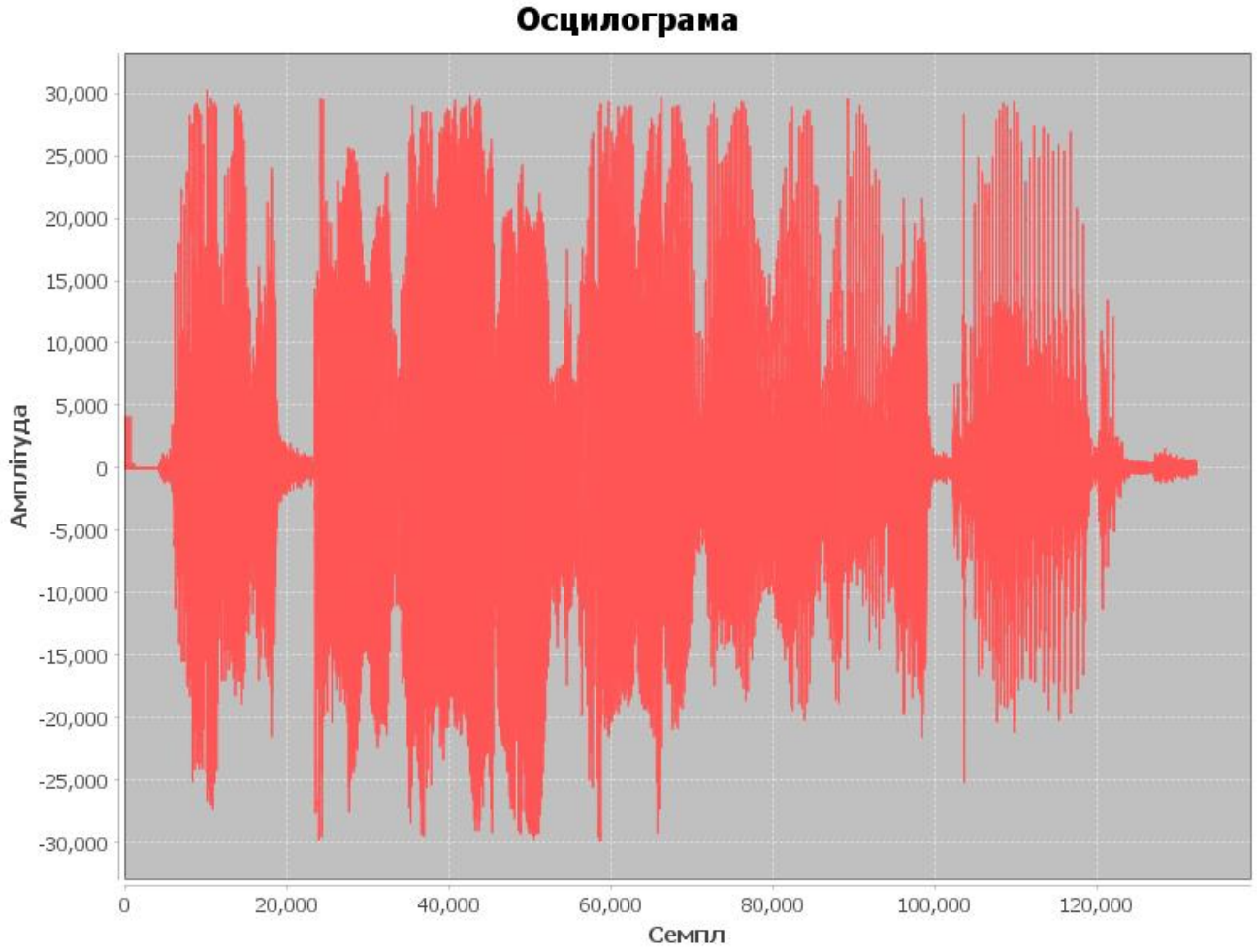

Рисунок 3.6 – Осцилограма аудіофайлу із ЦВЗ

Далі таким же чином була створена гістограма розподілу амплітуд, що показує, як часто зустрічаються різні значення амплітуд у звуковому сигналі. Це дає уявлення про загальний характер звуку, наприклад, музика має більш складний і різноманітний розподіл амплітуд, ніж простий тон. Високі стовпчики на гістограмі вказують на часті гучні звуки, а низькі стовпчики – на тихі звуки. Різкі піки або нерівномірності в гістограмі можуть свідчити про наявність шумів або спотворень у звуковому сигналі. За отриманими даними з гістограми складно визначити чи наявний в аудіофайлі ЦВЗ, але при його неправильному вбудовуванні було б одразу помітно велику різницю між наведеними амплітудами. Результат виконання даного етапу перевірки наведено на рисунку 3.7 та на рисунку 3.8.

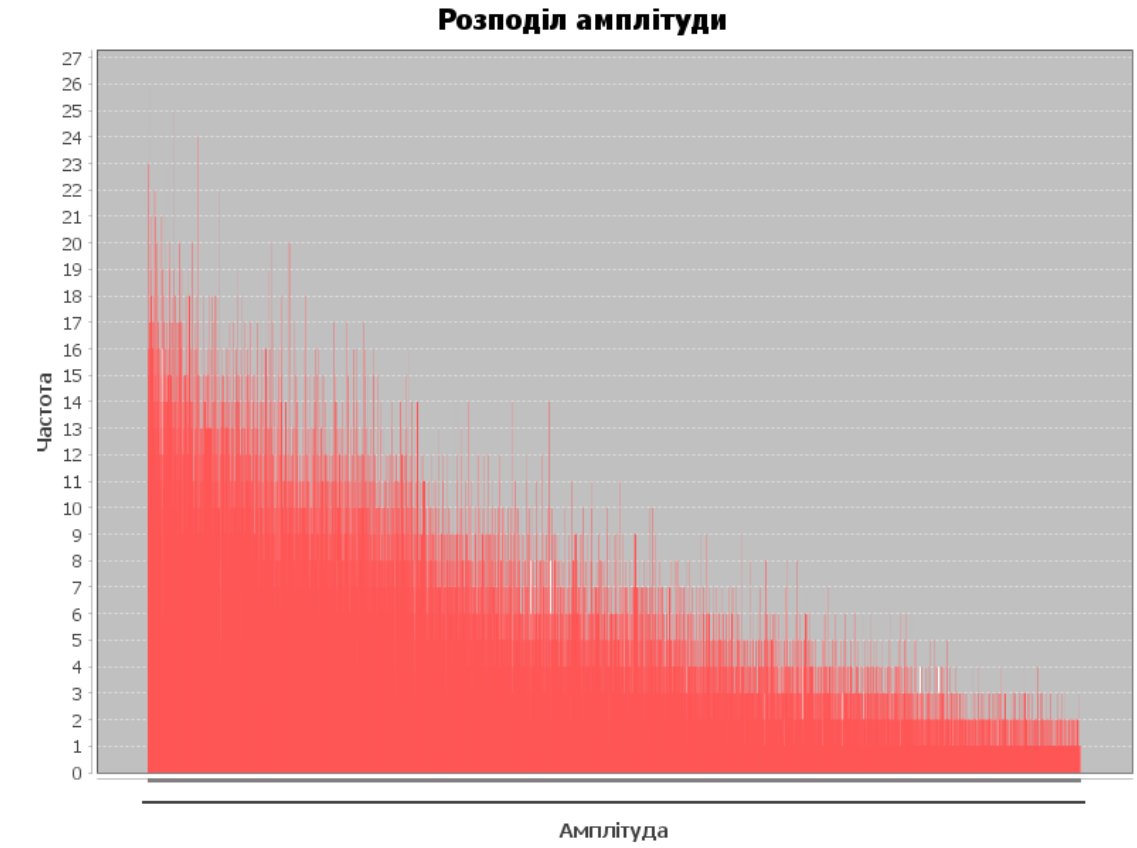

Рисунок 3.7 – Гістограма оригінального аудіофайлу

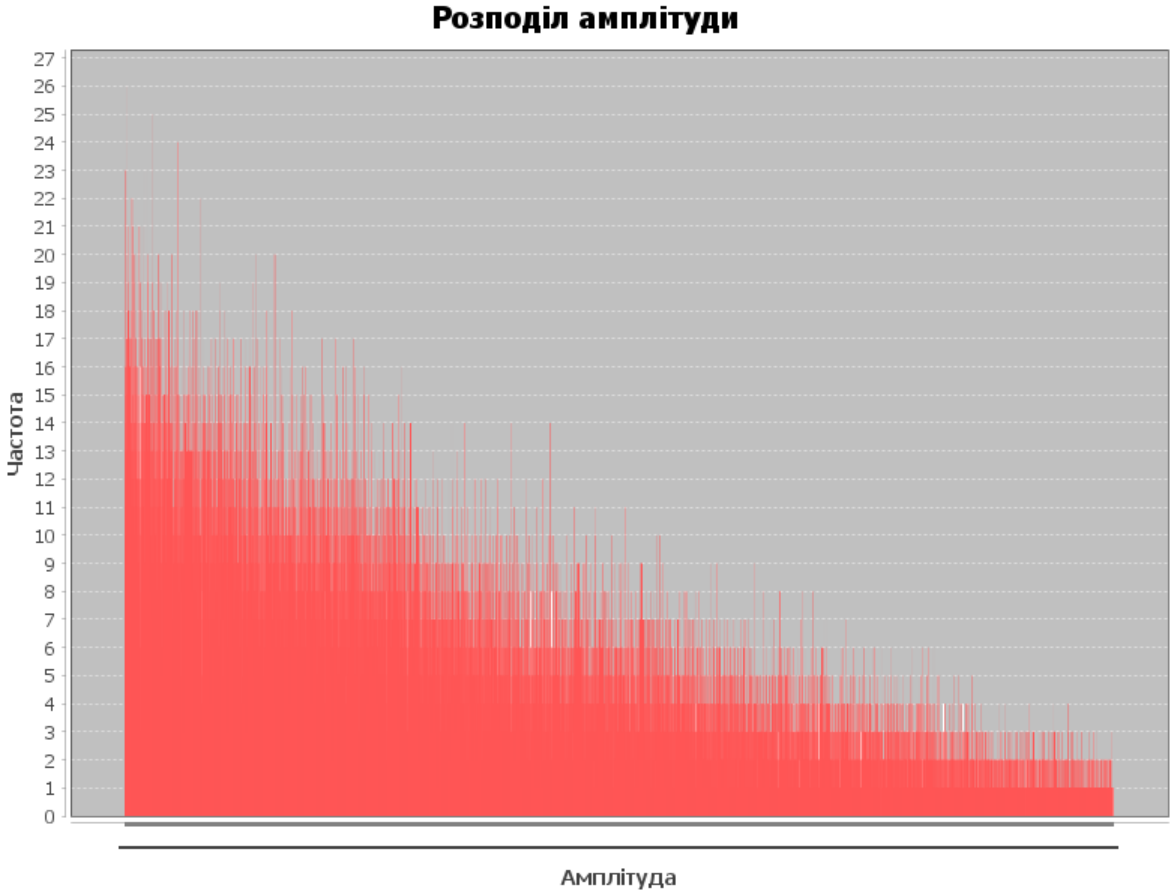

Рисунок 3.8 – Гістограма аудіофайлу із ЦВЗ

ВИСНОВОК

Метою даної дипломної роботи була розробка та впровадження методики адаптивного вбудовування цифрових водяних знаків в аудіофайли. В процесі дослідження були досягнуті наступні результати відповідно до встановлених задач на початку роботи. Було досліджено принципи стеганографії, зокрема у контексті GSM мереж, з метою забезпечення автентифікації обох сторін встановленого зв’язку. Це включало вивчення існуючих методик вбудовування цифрових водяних знаків та використання методів прямого розширення спектру сигналу. Було проаналізовано різноманітні методи розширення спектру та визначено їх особливості і недоліки. Цей аналіз включав порівняння різних підходів та обґрунтування вибору найбільш ефективного методу для даного дослідження.

Після чого було досліджено та обрано основну методику для вбудування цифрових водяних знаків в аудіофайли на основі послідовностей Голда. Вони були обрані через їх здатність утворювати квазіортогональний сигнал, що дозволяє однозначно ідентифікувати користувачів під час встановлення зв’язку та ускладнює можливість компрометації водяних знаків. Проведено детальний аналіз послідовностей Голда з метою оцінки їх стійкості щодо різноманітних атак з боку зловмисників. Це дозволило оцінити їх придатність для однозначної ідентифікації абонентів, особливо в умовах використання штучного інтелекту та створення недостовірної інформації.

Практична реалізація методики була виконана за допомогою мови програмування Java. На основі аудіофайлу та текстового файлу було програмно реалізоване середовище для їх обробки та аналізу. Проведений детальний аналіз утвореного сигналу з водяним знаком і без нього, вказує на те, що вбудування цифрових водяних знаків на основі кодів Голда має великий потенціал, через те що в такому випадку отримується основна сутність стеганографії, а саме передавання інформації відкритим джерелом, без явної її компрометації для третіх осіб. Тобто це дослідження вказало на доцільність використання даної методики в реальних пристроях. Загалом, виконана робота підтвердила можливість використання розробленої методики для забезпечення автентифікації та підвищення безпеки в

мережах зв’язку. Вбудовування цифрових водяних знаків в аудіофайли за допомогою послідовностей Голда демонструє високий рівень стійкості до атак та ефективність у створенні надійного засобу ідентифікації.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The analysis on the direct sequence spread spectrum communication system. SPIE Digital library : веб-сайт. Режим доступу: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12175/121750C/The-analysis-on-the-direct-sequence-spread-spectrum-communication-system/10.1117/12.2628421.full#_=_ (Дата звернення: 26.04.2024).

2. Don Torrieri. Principles of Spread-Spectrum Communication Systems : Книга. США : Університетська книга, 2018. 727 с. Режим доступу: https://link.springer.com/book/10.1007/978-3-319-70569-9 (Дата звернення: 29.04.2024).

3. Vladimir B. R., Branislav M. T., Nenad M. S. Frequency hopping spread spectrum: History, principles and applications. 2022. №4. Режим доступу: https://www.redalyc.org/journal/6617/661773214005/html/(Дата звернення: 29.04.2024)

4. M. Katta Swamy, M.Deepthi, V.Mounika, R.N.Saranya. Performance Analysis of DSSS and FHSS Techniques over AWGN Channel. 2023. №1. Режим доступу: https://www.longdom.org/open-access-pdfs/performance-analysis-of-dsss-and-fhss-techniques-over-awgn-channel-0976-4860-4-59-65.pdfДата звернення: 02.05.2024)

5. Quan Wang. Non-linear chirp spread spectrum communication systems of binary orthogonal keying mode : Книга. Канада : Університетська книга, 2015. 211 с. Режим доступу: https://ir.lib.uwo.ca/cgi/viewcontent.cgi?article=4330&context=etd (Дата звернення: 04.05.2024).

6. Lisa M. Marvel, Charles G. Boncelet, Jr., Charles T. Retter. Methodology of Spread-Spectrum Image Steganography mode : Книга. США : Університетська книга, 1998. 37 с. Режим доступу: https://apps.dtic.mil/sti/tr/pdf/ADA349102.pdf (Дата звернення: 06.05.2024).

7. Kuznetsov, O., Frontoni, E. & Chernov, K. Beyond traditional steganography: enhancing security and performance with spread spectrum image steganography. 2024. №54. Режим доступу:

https://link.springer.com/article/10.1007/s10489-024-05415-z (Дата звернення: 08.05.2024)

8. Mark W. Young. Design of a Gold Code Generator for Use in Code Division Multiple Access Communication System : Книга. США : Університетська книга, 1985. 135 с. Режим доступу: https://core.ac.uk/download/pdf/236303925.pdf (Дата звернення: 10.05.2024).

9. Mark W. Young. Design of a Gold Code Generator for Use in Code Division Multiple Access Communication System : Книга. США : Університетська книга, 1985. 135 с. Режим доступу: https://core.ac.uk/download/pdf/236303925.pdf (Дата звернення: 10.05.2024).

10. S.E. El-khamyj, A.S. Balamesh, B. H. Morfequ. On the generation and correlation measurements of Gold codes : Книга. США : Університетська книга, 1987. 492 с. Режим доступу: https://www.tandfonline.com/doi/pdf/10.1080/00207218808962821#:~:text=The%20Gold%20code%20is%20generated,Gold%20Code%20on%20an%20oscilloscope. (Дата звернення: 11.05.2024).

11. Конаховіч Г.Ф., Пузирєнко А.Ю. Комп’ютерна стеганографія : Книга. Україна : Університетська книга, 2006. 288 с. Режим доступу: https://studfile.net/preview/7379018/. (Дата звернення: 13.05.2024).

ДОДАТОК А

```
package org.example;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartUtils;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.DefaultCategoryDataset;
import org.jfree.data.xy.XYSeries;
import org.jfree.data.xy.XYSeriesCollection;
import javax.sound.sampled.*;
import java.io.*;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.util.BitSet;

public class Main {
    static int interval = 100;

    public static void main(String[] args) throws
UnsupportedAudioFileException, IOException {
        String audioFile = "1.wav"; //Оригінальний аудіофайл
        String watermarkFile = "1.txt"; //Цифровий водяний знак
        String watermarkedFile = "watermarked.wav"; //Вихідний аудіофайл з
водяним знаком, що буде створений в процесі

        //1. Завантаження аудіофайлу та текстового файлу у вигляді
послідовності байт
        AudioInputStream ais = AudioSystem.getAudioInputStream(new
File(audioFile));
        byte[] audioBytes = ais.readAllBytes();
        BitSet audioData = BitSet.valueOf(audioBytes);
        String watermarkText = new String(Files.readAllBytes(new
File(watermarkFile).toPath()));
        byte[] watermarkBytes = watermarkText.getBytes();
        BitSet watermarkData = BitSet.valueOf(watermarkBytes);
        int watermarkLength = watermarkData.length();

        //2. Генерація послідовності кодів Голда
        BitSet goldCodes = generateGoldCodes(watermarkData.length());
```

```
        //3. Вбудування підпису за допомогою кодів Голда
        for (int i = 0; i < watermarkData.length(); i++) {
            int audioIndex = i * interval; //Інтервал вставлення бітів водяного знаку, тобто через кожні interval біт даних аудіофайлу буде вставлятись 1 біт цифрого знаку
            if (watermarkData.get(i)) { //Вставлення біта в аудіофайл audioData по індексу audioIndex в значення, що береться з послідовності Голда goldCodes по індексу i
                audioData.set(audioIndex, goldCodes.get(i));
            } else {
                audioData.clear(audioIndex);
            }
        }

        //4. Конвертація з набору даних в підписаний аудіофайл та його збереження
        byte[] watermarkedBytes = audioData.toByteArray();
        AudioFormat format = ais.getFormat();
        AudioInputStream watermarkedAIS = new AudioInputStream(new ByteArrayInputStream(watermarkedBytes), format, watermarkedBytes.length / format.getFrameSize());
        AudioSystem.write(watermarkedAIS, AudioFileFormat.Type.WAVE, new File(watermarkedFile));
        System.out.println("Цифровий водяний знак успішно вбудовано в аудіофайл та збережено у файлі " + watermarkedFile);

        //5. Аналіз шляхом видобування вбудованого ЦВЗ

System.out.println("\n|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||");
        System.out.println("1) За допомогою зворотнього алгоритму вбудовування, видобувається ЦВЗ з аудіофайлу" +
                "\nОригінальний текст ЦВЗ: " + watermarkText);
        extractAndDisplayWatermark(watermarkedFile, watermarkLength, goldCodes);

System.out.println("||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||");

        //6. Аналіз даних шляхом ітеративного перебору всіх бітів з двох аудіофайлів
```

```
System.out.println("\n|||||||||||||||||||||||||||||||||||||||||||||||||||||||||
|||||||");
        System.out.println("2) Аналіз даних шляхом ітеративного перебору всіх
бітів з двох аудіофайлів," +
                "\nнижче виведено саме ті біти які відрізняються між двома
аудіофайлами," +
                "\n що доводить успішне вбудування ЦВЗ");
        compareBitByBit(audioFile, watermarkedFile);
        System.out.println("\n2.1) Додаткова перевірка шляхом порівняння між
собою двох однакових файлів за допомогою того ж методу");
        System.out.println("Порівняння двох однакових аудіофайлів без ЦВЗ");
        compareBitByBit(audioFile, audioFile);
        System.out.println("\nПорівняння двох однакових аудіофайлів із ЦВЗ");
        compareBitByBit(watermarkedFile, watermarkedFile);

System.out.println("|||||||||||||||||||||||||||||||||||||||||||||||||||||||||
|||||");

        //7. Кореляційний аналіз

System.out.println("\n|||||||||||||||||||||||||||||||||||||||||||||||||||||||||
|||||||");
        System.out.println("3) Проведення кореляційного аналізу між двома
наборами даних" +
                "\n(аудіофайл з ЦВЗ та без нього) з допустимим порогом 0.8");
        verifyWatermarkByCorrelation(audioData, watermarkedFile);

System.out.println("|||||||||||||||||||||||||||||||||||||||||||||||||||||||||
|||||");

        //8. Аналіз властивостей аудіофайлів

System.out.println("\n|||||||||||||||||||||||||||||||||||||||||||||||||||||||||
|||||||");
        System.out.println("4) Аналіз даних аудіофайлів");
        analyzeAudio(audioFile);
        System.out.println();
        analyzeAudio(watermarkedFile);

System.out.println("|||||||||||||||||||||||||||||||||||||||||||||||||||||||||
|||||");
```

```
        //9. Візуальний аналіз

System.out.println("\n|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
|||||||");
        System.out.println("5) Побудова графіків сигналу та гістограм");
        int[] originalSamples = loadAudioSamples(audioFile);
        int[] watermarkedSamples = loadAudioSamples(watermarkedFile);
        plotAndSaveWaveform(originalSamples, "original_waveform.png");
        plotAndSaveWaveform(watermarkedSamples, "watermarked_waveform.png");
        plotAndSaveHistogram(originalSamples, "original_histogram.png");
        plotAndSaveHistogram(watermarkedSamples,
"watermarked_histogram.png");
    }

    //Генерація кодів Голда шляхом використання регістру зсуву зі зворотнюю
сумісністю LFSR
    private static BitSet generateGoldCodes(int length) {
        int mSeqLength = 31; //Довжина максимальної m-послідовності LFSR
        int polynomial1 = 0x25; //x^5 + x^3 + x^2 + x^1 + 1
        int polynomial2 = 0x27; //x^5 + x^4 + x^2 + x^1 + 1
        BitSet mSeq1 = generateMSeq(mSeqLength, polynomial1); //Генерація m-
послідовності у вигляді набору бітів, з використанням параметру довжини та
полінома
        BitSet mSeq2 = generateMSeq(mSeqLength, polynomial2);
        BitSet goldCodes = new BitSet(length);
        for (int i = 0; i < length; i++) {
            goldCodes.set(i, mSeq1.get(i) ^ mSeq2.get((i + 10) % mSeqLength));
//Утворення кодів Голда шляхом встановлення і біта в послідовність goldCodes,
об'єднання першої та другої послідовності mSeq1 та mSeq2 операцією XOR, при
цьому mSeq2 здвигається на 10 позицій, після оператором остатку від ділення %
забезпечується приведення до довжини послідовності і відповідно циклічності
операцій
        }
        return goldCodes;
    }

    //Генерація m-послідовності
    private static BitSet generateMSeq(int length, int polynomial) {
        BitSet mSeq = new BitSet(length);
        mSeq.set(0); //Початковий стан регістру
        for (int i = 1; i < length; i++) {
            boolean feedback = mSeq.get(i - 1); //Отримання попереднього біта
з послідовності(0 або 1)
```

```
            for (int j = 1; j < 32; j++) {
                if (i - j >= 0 && ((polynomial >> j) & 1) == 1) { //Попередження звертання до від'ємних бітів та подальша перевірка чи встановлено біт на позиції j в поліномі, що дає змогу чи необхідно використовувати значення feedback
                    feedback ^= mSeq.get(i - j); //Виконання операції XOR між попереднім бітом і бітом з попереднього відведення
                }
            }
            mSeq.set(i, feedback); //Утворення послідовності
        }
        return mSeq;
    }

    //Видобування цифрового водяного знаку з аудіофайлу та його демонстрація
    private static void extractAndDisplayWatermark(String watermarkedFile, int watermarkLength, BitSet goldCodes) throws UnsupportedAudioFileException, IOException {
        AudioInputStream watermarkedAIS = AudioSystem.getAudioInputStream(new File(watermarkedFile));
        byte[] watermarkedBytes = watermarkedAIS.readAllBytes();
        BitSet watermarkedData = BitSet.valueOf(watermarkedBytes).get(0, watermarkLength);
        BitSet extractedWatermark = new BitSet(watermarkLength);
        for (int i = 0; i < watermarkLength; i++) {
            if (watermarkedData.get(i) == goldCodes.get(i)) {
                extractedWatermark.set(i);
            } else {
                extractedWatermark.clear(i);
            }
        }
        byte[] watermarkBytes = extractedWatermark.toByteArray();
        String extractedText = new String(watermarkBytes, StandardCharsets.UTF_8);
        System.out.println("Видобутий водяний цифровий знак: " + extractedText);
    }

    //Побітове порівняння
    private static void compareBitByBit(String originalFile, String watermarkedFile) throws UnsupportedAudioFileException, IOException {
        AudioInputStream originalAIS = AudioSystem.getAudioInputStream(new File(originalFile));
```

```
        byte[] originalBytes = originalAIS.readAllBytes();
        BitSet originalData = BitSet.valueOf(originalBytes);
        AudioInputStream watermarkedAIS = AudioSystem.getAudioInputStream(new
File(watermarkedFile));
        byte[] watermarkedBytes = watermarkedAIS.readAllBytes();
        BitSet watermarkedData = BitSet.valueOf(watermarkedBytes);
        boolean watermarkDetected = false;
        for (int i = 0; i < originalData.length(); i++) {
            if (originalData.get(i) != watermarkedData.get(i)) {
                watermarkDetected = true;
                break;
            }
        }
        if (watermarkDetected) {
            System.out.println("РЕЗУЛЬТАТ ПОБІТОВОЇ ПЕРЕВІРКИ: Виявлено
різницю в бітах файлів!");
        } else {
            System.out.println("РЕЗУЛЬТАТ ПОБІТОВОЇ ПЕРЕВІРКИ: Файли
однакові, різниці між бітами файлів немає!");
        }
    }

    //
    private static void verifyWatermarkByCorrelation(BitSet audioData, String
watermarkedFile) throws UnsupportedAudioFileException, IOException {
        double correlationThreshold = 0.8; //Поріг кореляції
        AudioInputStream watermarkedAIS = AudioSystem.getAudioInputStream(new
File(watermarkedFile));
        byte[] watermarkedBytes = watermarkedAIS.readAllBytes();
        BitSet watermarkedData = BitSet.valueOf(watermarkedBytes);
        double correlation = calculateCorrelation(watermarkedData,
audioData);
        if (correlation >= correlationThreshold) {
            System.out.println("Значення кореляції в допустимому діапазоні: "
+ correlation + ").");
        } else {
            System.out.println("Значення кореляції в недопустимому діапазоні:
" + correlation + ").");
        }
    }

    //Метод розрахунку коефіцієнта кореляції Пірсона
    private static double calculateCorrelation(BitSet data1, BitSet data2) {
```

```
        if (data1.length() != data2.length()) {
            throw new IllegalArgumentException("Набори бітів мають бути однакової довжини!");
        }
        int n = data1.length();
        double sumXY = 0, sumX = 0, sumY = 0, sumX2 = 0, sumY2 = 0;
        for (int i = 0; i < n; i++) {
            double x = data1.get(i) ? 1 : 0;
            double y = data2.get(i) ? 1 : 0;
            sumXY += x * y;
            sumX += x;
            sumY += y;
            sumX2 += x * x;
            sumY2 += y * y;
        }
        double numerator = n * sumXY - sumX * sumY;
        double denominator = Math.sqrt((n * sumX2 - sumX * sumX) * (n * sumY2 - sumY * sumY));
        if (denominator == 0) {
            return 0;
        }
        return numerator / denominator;
    }

    private static void analyzeAudio(String audioFile) throws UnsupportedAudioFileException, IOException {
        AudioInputStream ais = AudioSystem.getAudioInputStream(new File(audioFile));
        AudioFormat format = ais.getFormat();
        byte[] audioBytes = ais.readAllBytes();
        int numChannels = format.getChannels();
        int sampleSizeInBits = format.getSampleSizeInBits();
        int[] samples = convertToSamples(audioBytes, format.getSampleSizeInBits());
        double mean = calculateMean(samples);
        double variance = calculateVariance(samples, mean);
        double stdDeviation = Math.sqrt(variance);
        double skewness = calculateSkewness(samples, mean, stdDeviation);
        double kurtosis = calculateKurtosis(samples, mean, stdDeviation);
        System.out.println("Назва файлу: " + audioFile);
        System.out.println("Кількість каналів звуку: " + numChannels);
        System.out.println("Розмір семплу в бітах: " + sampleSizeInBits);
```

```
        System.out.println("Частота дискретизації: " +
format.getSampleRate());
        System.out.println("Середнє значення амплітуд семплів: " + mean);
        System.out.println("Дисперсія: " + variance);
        System.out.println("Стандартне відхилення: " + stdDeviation);
        System.out.println("Асиметрія, форма розподілу амплітуд: " +
skewness);
        System.out.println("Ексцес: " + kurtosis);
    }

    //Середнє значення амплітуд семплів
    private static double calculateMean(int[] data) {
        long sum = 0;
        for (int value : data) {
            sum += value;
        }
        return (double) sum / data.length;
    }

    //Дисперсія
    private static double calculateVariance(int[] data, double mean) {
        double sumSquaredDeviations = 0;
        for (int value : data) {
            double deviation = value - mean;
            sumSquaredDeviations += deviation * deviation;
        }
        return sumSquaredDeviations / data.length;
    }

    //Асиметрія
    private static double calculateSkewness(int[] data, double mean, double
stdDeviation) {
        double sumCubedDeviations = 0;
        for (int value : data) {
            double deviation = value - mean;
            sumCubedDeviations += deviation * deviation * deviation;
        }
        return sumCubedDeviations / (data.length * stdDeviation * stdDeviation
* stdDeviation);
    }

    //Ексцес
```

```
    private static double calculateKurtosis(int[] data, double mean, double
stdDeviation) {
        double sumFourthPowerDeviations = 0;
        for (int value : data) {
            double deviation = value - mean;
            sumFourthPowerDeviations += deviation * deviation * deviation *
deviation;
        }
        return sumFourthPowerDeviations / (data.length * stdDeviation *
stdDeviation * stdDeviation * stdDeviation) - 3;
    }

    //Завантаження аудіофайлу
    private static int[] loadAudioSamples(String filePath) throws
UnsupportedAudioFileException, IOException {
        AudioInputStream ais = AudioSystem.getAudioInputStream(new
File(filePath));
        AudioFormat format = ais.getFormat();
        byte[] audioBytes = ais.readAllBytes();
        return convertToSamples(audioBytes, format.getSampleSizeInBits());
    }

    //Перетворення масиву байтів в масив семплів
    private static int[] convertToSamples(byte[] audioBytes, int
sampleSizeInBits) {
        int[] samples = new int[audioBytes.length / (sampleSizeInBits / 8)];
        ByteBuffer buffer =
ByteBuffer.wrap(audioBytes).order(ByteOrder.LITTLE_ENDIAN);
        for (int i = 0; i < samples.length; i++) {
            if (sampleSizeInBits == 16) {
                samples[i] = buffer.getShort();
            } else if (sampleSizeInBits == 8) {
                samples[i] = buffer.get();
            }
        }
        return samples;
    }

    //Побудова осцилограми звукового файлу
    private static void plotAndSaveWaveform(int[] samples, String filename)
throws IOException {
        XYSeries series = new XYSeries("Осцилограма");
        for (int i = 0; i < samples.length; i++) {
```

```
            series.add(i, samples[i]);
        }
        XYSeriesCollection dataset = new XYSeriesCollection(series);
        JFreeChart chart = ChartFactory.createXYLineChart("Осцилограма",
"Семпл", "Амплітуда", dataset, PlotOrientation.VERTICAL, false, false, false);
        ChartUtils.saveChartAsPNG(new File(filename), chart, 800, 600);
    }

    //Побудова гістограми розподілу амплітуди
    private static void plotAndSaveHistogram(int[] samples, String filename)
throws IOException {
        int[] histogram = calculateHistogram(samples);
        DefaultCategoryDataset dataset = new DefaultCategoryDataset();
        for (int i = 1000; i < histogram.length; i++) {
            if (histogram[i] > 0) {
                dataset.addValue(histogram[i], "Частота", String.valueOf(i));
            }
        }
        JFreeChart chart = ChartFactory.createBarChart("Розподіл амплітуди",
"Амплітуда", "Частота", dataset, PlotOrientation.VERTICAL, false, false,
false);
        ChartUtils.saveChartAsPNG(new File(filename), chart, 800, 600);
    }

    //Розрахунок даних для гістограми
    private static int[] calculateHistogram(int[] data) {
        int maxAmplitude = 0;
        for (int sample : data) {
            maxAmplitude = Math.max(maxAmplitude, Math.abs(sample));
        }

        int[] histogram = new int[maxAmplitude + 1];
        for (int sample : data) {
            histogram[Math.abs(sample)]++;
        }
        return histogram;
    }
}
```