Вінницький національний технічний університет
Факультет інформаційних електронних систем
Кафедра інфокомунікаційних систем і технологій

Бакалаврська кваліфікаційна робота на тему:

ДОСЛІДЖЕННЯ АЛГОРИТМІВ ЗАХИСТУ АУДІО ІНФОРМАЦІЇ

Виконав: студент 4-го курсу, групи ПЗТ-21б
спеціальності 172 –Телекомунікації та радіотехніка
Коновченко Д.О.

Керівник: к.т.н., доц., доцент каф. ІКСТ
Стальченко О.В.
«12» червня 2025 р.

Рецензент: д.т.н., проф., професор каф. ІРТС
Семенов А.О.
«12» червня 2025 р.

Допущено до захисту

Завідувач кафедри ІКСТ

д.т.н., проф. Кичак В.М.

«12» червня 2025 р.

Вінниця ВНТУ - 2025 рік

Вінницький національний технічний університет
Факультет інформаційних електронних систем
Кафедра інфокомунікаційних систем та технологій
Рівень вищої освіти перший (бакалаврський)
Галузь знань 17 – Електроніка та телекомунікації
(шифр і назва)
Спеціальність 172 – Телекомунікації та радіотехніка
(шифр і назва)
Освітня програма – Програмне забезпечення телекомунікаційних систем

ЗАТВЕРДЖУЮ
Завідувач кафедри ІКСТ
д.т.н., професор В.М. Кичак
«05» 05 2025 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧЦІ

Коновченко Дар'ї Олександрівні
(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження алгоритмів захисту аудіо інформації
керівник роботи Стальченко Олександр Володимирович, канд. техн. наук,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом ВНТУ від "20" березня 2025 року № 97

2. Строк подання здобувачем роботи: 09 червня 2025 року

3. Вихідні дані до роботи: тип вхідних сигналів – аудіо сигнали; режим зв'язку – дуплексний; смуга робочих частот – 300-3400 Гц; тип модуляції – ДМ; тип кодування – комбінований.

4. Зміст розрахунково-пояснювальної записки роботи (перелік питань, які потрібно розробити): вступ, огляд методів захисту аудіо інформації, стиснення мовних сигналів на основі дельта-перетворення другого порядку, розробка алгоритму захисту з використанням модифікації алгоритму Rijndael, алгоритм захисту голосових повідомлень, алгоритм А2, охорона праці.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): структурна схема кодування; графіки дельта-перетворення; структури регістрів зсуву; результати моделювання розроблених алгоритмів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Спеціальна частина	Стальченко О.В., доцент кафедри ІКСТ	05.05.2025	12.05.2025
Охорона праці	Дембіцька С.В., професор кафедри БЖДПБ	05.05.20	12.05.2025

7. Дата видачі завдання «05» травня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Формування та затвердження теми та індивідуального завдання бакалаврської кваліфікаційної роботи (БКР)	06.05.2025р.	
2	Виконання спеціальної частини БКР. Перший рубіжний контроль виконання БКР	19.05.2025р.	
3	Виконання спеціальної частини БКР. Другий рубіжний контроль виконання БКР	02.06.2025р.	
4	Виконання розділу «Охорона праці»	06.06.2025р.	
5	Нормоконтроль БКР	09.06.2025р.	
6	Попередній захист БКР	10.06.2025р.	
7	Рецензування БКР	11.06.2025р.	
8	Захист БКР	13.06.2025р.	

Здобувачка _____ (підпис) Коновченко Д.О.

Керівник роботи _____ (підпис) Стальченко О.В.

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 114 сторінок формату А4, на яких є 13 рисунків, 10 таблиць, список використаних джерел містить 25 найменувань.

Метою даної бакалаврської роботи є розробка алгоритмів криптографічного захисту аудіо інформації на основі стиснення з використанням оптимізованих дельта-перетворень другого порядку, які є перспективними для вирішення даної задачі з точки зору властивостей дельта-послідовностей, швидкості та простоти алгоритмів стиснення.

У даній бакалаврській роботі було проведено синтез та дослідження основних алгоритмів захисту аудіо інформації із використанням дельта перетворення та блокових шифрів.

Ключові слова: захист інформації, аудіо інформація, дельта перетворення, блочні шифри

ABSTRACT

The bachelor's qualification work consists of 114 pages of A4 format, which include 13 figures, 10 tables, and the list of sources used contains 25 items.

The purpose of this bachelor's work is to develop algorithms for cryptographic protection of audio information based on compression using optimized second-order delta transforms, which are promising for solving this problem from the point of view of the properties of delta sequences, speed and simplicity of compression algorithms.

In this bachelor's work, a synthesis and study of the main algorithms for protecting audio information using delta transforms and block ciphers was carried out.

Keywords: information protection, audio information, delta transforms, block ciphers.

Зміст

ВСТУП

Актуальність. Широке використання інформаційних технологій та засоби зв'язку в автоматизованих системах обробки і контролю інформації ставлять завдання розробки принципово нових методів обробки, зберігання і захисту інформації. Це зробило актуальною і гострою проблему захисту інформації, що передається каналами зв'язку загального користування.

За оцінками вітчизняних і зарубіжних фахівців, значну частку інформації, що передається по каналах зв'язку загального користування, складають голосові повідомлення, тому завдання захисту мовної інформації займають одне з провідних місць у вирішенні загальної проблеми інформаційної безпеки [1, 2]. Це пов'язано з тим, що мова є універсальним інструментом людського спілкування, який має унікальні особливості ефекту присутності, емоційного забарвлення, аутентифікації, надмірності інформації, і в багатьох системах комунікації та передачі інформації йому важко знайти якусь рівноцінну заміну.

Аналіз останніх досліджень. Відповідно до цього голосове повідомлення слід розглядати як особливий вид інформаційного повідомлення зі специфічними характеристиками, що визначають застосування спеціальних заходів захисту для вирішення таких завдань: запобігання витоку голосової інформації внаслідок несанкціонованого прослуховування в телекомунікаційних каналах; Попередження змін у мовленнєвому повідомленні шляхом модифікації значення мовних фраз або особистісних характеристик мовця.

Сучасні досягнення наукової і технічної думки в галузі інформаційних технологій змусили нас по-новому поглянути на проблему безпеки мовного спілкування. Вражаючі досягнення в області комп'ютерної та стільникової телефонії, розробка мультимедійних засобів і методів передачі аудіо-відео інформації по комп'ютерних мережах (IP-телефонія, відеоконференцзв'язок) підштовхують розробників технологій до необхідності створення загального алгоритмічного підходу до побудови і моделювання спеціалізованих алгоритмів

перетворення мови, який досить легко реалізується програмним забезпеченням на основі стандартної комп'ютерної технології.

Для голосових повідомлень в даний час існує 2 найбільш поширені матеріальні форми - аналогове і цифрове представлення голосових сигналів, що зумовило використання в якості технологій безпеки голосового зв'язку у відкритих каналах двох підходів: зашифровки голосових повідомлень і криптографічного захисту. Перший підхід традиційно використовується для аналогових сигналів, а другий - для дискретних повідомлень. Використання методів скремблювання для захисту мовних повідомлень в цифровому представленні не має широкого поширення, так як захисні скремблінгові перетворення для цифрових мовних сигналів характеризуються високою ресурсною інтенсивністю, а рівень залишкової розбірливості таких методів досить високий.

Перехід від аналогових методів обробки сигналів до цифрових методів обробки вимагає розробки нових методів захисту цифрових мовних сигналів [2,3]. У зв'язку з розвитком технологій зв'язку і широким використанням цифрової техніки саме криптографічний захист голосових сигналів представляє найбільший інтерес, так як вона забезпечує більшу стабільність в порівнянні з скремблінгом.

В даний час для захисту мовної інформації використовуються методи дискретної криптографії, які в більшості своїй не враховують специфіку формування мовних повідомлень і не відповідають вимогам до продуктивності більшості сучасних додатків. Це є проявом сучасної тенденції використання уніфікованих алгоритмів захисту інформації на шкоду розвитку окремі засоби захисту окремих типів повідомлень, що забезпечують захист повідомлень даного типу та з урахуванням специфіки їх формування за джерелом.

Оскільки голосові повідомлення в цифровому вигляді займають значний обсяг, використання досить трудомістких процедур шифрування існуючих блокових шифрів для захисту цифрових голосових повідомлень може виявитися недоцільним з міркувань обчислювальних зусиль. Для зниження трудомісткості

і збільшення швидкості виконання процедур захисту доцільно використовувати меншу кількість раундів криптографічних перетворень, але при зменшенні числа патронів необхідно забезпечити прийнятну міцність. Одним із шляхів вирішення цієї проблеми є використання високошвидкісних алгоритмів стиснення голосових повідомлень для зменшення обсягу інформації та побудови на їх основі параметризованих алгоритмів захисту з урахуванням інформаційних характеристик стисненого потоку для створення криптоаналітика великої невизначеності при вирішенні завдання криптоаналізу.

Таким чином, дослідження, спрямоване на розробку алгоритмів захисту мовленнєвих повідомлень в рамках вирішення зазначених вище завдань, є актуальним і становить науковий і практичний інтерес.

Оптимізовані дельта-перетворення другого порядку розглядаються як перспективні методи стиснення мовних повідомлень з точки зору захисту інформації, а алгоритми захисту мовної інформації запропоновані на основі властивостей дельта-послідовностей.

Дана робота присвячена проблемі побудови ефективних алгоритмів криптографічного захисту голосових повідомлень в реальному часі на основі стиснення з використанням оптимізованих дельта-перетворень другого порядку.

Об'єкт дослідження

Методи та алгоритми криптографічного захисту інформації, алгоритми стиснення голосового сигналу.

Мета і завдання роботи

Метою даної бакалаврської роботи є розробка алгоритмів криптографічного захисту аудіо інформації на основі стиснення з використанням оптимізованих дельта-перетворень другого порядку, які є перспективними для вирішення даної задачі з точки зору властивостей дельта-послідовностей, швидкості та простоти алгоритмів стиснення.

Для досягнення поставленої мети в роботі розглядаються такі основні задачі:

1. Аналіз існуючих методів і алгоритмів захисту мовних повідомлень,

виявлення переваг і недоліків цих методів;

2. Аналіз алгоритмів стиснення мовних повідомлень;

3. Розгляд алгоритмів оптимізованих дельта-перетворень для стиснення мовної інформації та аналіз властивостей дельта-послідовностей;

4. Розробка алгоритмів захисту і стиснених голосових повідомлень на основі оптимізованих дельта-перетворень другого порядку

Методи дослідження. У ході даної роботи були використані, елементи теорії інформації для дослідження методів кодування, математичної логіки для побудови кодів, теорії оптимізованих дельта-перетворень для перетворення аудіо інформації, елементи математичної статистики порівняння результатів досліджень, а також математичний апарат модульної арифметики в скінченних полях. В якості основних методів дослідження були використані математичні моделі та імітаційне комп'ютерне моделювання.

Апробація роботи. Основні ідеї роботи доповідались і обговорювались на науковій конференції ВНТУ у 2025 році.

1. ОГЛЯД МЕТОДІВ ЗАХИСТУ АУДІО ІНФОРМАЦІЇ

В даний час існує значна невизначеність базових понять в області інформаційної безпеки, в зв'язку з чим необхідно ввести класифікацію основних понять і визначень, що використовуються в даній дисертації.

Початковим поняттям пропонується вважати поняття звукової інформації, яке визначається як будь-яка інформація, що сприймається слуховим апаратом людини. Основною складовою поняття аудіоінформації є мовна інформація - звукова інформація, що формується мовним апаратом людини.

У зв'язку з цим будемо вважати, що існує 2 форми подачі звукової інформації - логічна і матеріальна. Логічною формою представлення є звукове повідомлення (мовне повідомлення), яке має деякі інформаційні характеристики. Матеріальною формою представлення є сигнал (в даній роботі розглядаються цифрові і аналогові сигнали).

1.1. Методи захисту аналогових голосових повідомлень

Історично склалося так, що всі методи захисту аналогових мовних повідомлень об'єднані одним терміном скремблінг, під яким розуміють змішане перетворення сигналу. Аналогові сигнали (мовні, біомедичні, відео- і телевізійні) безперервні і характеризуються власним спектром (сукупністю синусоїдальних компонентів (гармонік), еквівалентних сигналу). Тому одним з методів скремблювання є перетворення мовного сигналу в частотній області: інверсія спектра і частотна перестановка. Найбільш широко використовується метод перемикання інверсії за законом ключа, який полягає в застосуванні циклічної інверсії зі зміною несучої частоти сигналу і передачею частини спектра. Основними недоліками цього методу є мала кількість можливих несучих частот, завдяки яким вихідний сигнал може бути швидко відновлений шляхом їх перерахування, і неприпустимо висока залишкова розбірливість вихідного сигналу (суб'єктивна характеристика, пов'язана з індивідуальними

особливостями слухового сприйняття людини). Третій спосіб зміни сигналу в частотній області полягає в поділі спектра сигналу на N рівних піддіапазонів, які можна міняти місцями і інвертувати. Так як є N! можливих перестановок і 2^N варіантів інверсій, загальна кількість варіантів перетворення сигналу складе $N!*2^N$. Незважаючи на велику кількість варіантів перебору, саме мале число піддіапазонів є обмеженням даного методу, так як введення великої кількості піддіапазонів пов'язане з практичними труднощами і спотвореннями сигналу. Крім того, зазвичай більше 40% енергії сигналу (для мовних сигналів) зосереджено в перших двох піддіапазонах, відповідних першому форманту, що значно послаблює довговічність методу, так як необхідно знайти правильні положення перших двох піддіапазонів і перемістити їх в потрібні місця.

Скремблери з часовою перестановкою ділять аналоговий сигнал на рівні проміжки часу, які називаються кадрами, кожен з яких ділиться на фрагменти. Перетворення сигналу виконується шляхом перестановки сегментів у кожному кадрі та перестановки кадрів повідомлень. Важливим фактором є розумний вибір довжин кадрів і відрізків, так як сигнал не руйнується всередині відрізка, відрізки слід вибирати настільки короткі, щоб вони не містили цілих фрагментів повідомлення (одиниць комбінованої мови). З іншого боку, довжина відрізка серйозно впливає на якість звучання переданого сигналу, що пояснюється чисто технічними причинами (чим менше відрізок, тим нижче якість звуку) [3].

Крім вибору довжин рамок і відрізків, важливим параметром є сама перестановка, так як одні перестановки кращі за інші (з точки зору залишкової розбірливості), і необхідний критерій відбору «хороших» перестановок. Для вирішення проблеми вибору «хороших» перестановок зазвичай використовується експертна оцінка залишкової розбірливості, так як формалізувати критерій відбору «хороших» перестановок досить складно [3, 4]. Випадкова перестановка, яка з формальної точки зору досить «непогана» (за умови високого змішування), може мати високу розбірливість, що обумовлено здатністю людського мозку відновлювати пропущений сегмент i+1 при прослуховуванні зашифрованого сигналу з парою сусідніх сегментів *i, i+2*.

Найвищою довговічністю володіють композитні скремблери, принцип побудови яких полягає в спільному використанні перемикання інверсії, частотного і часового змішування в одній системі, але такі системи також можна проаналізувати.

Самостійним методом захисту аналогових сигналів є використання апаратних генераторів шуму, коли сигнал з виходу генератора підсумовується з сигналом, що захищається, а результат передається по каналу зв'язку. Одержувач повідомлення повинен мати аналогічний генератор для відновлення повідомлення. Основними недоліками цього методу є необхідність жорсткої синхронізації передавального і приймального генераторів, складність конструкції самих генераторів і низька якість відновленого сигналу.

Слід зазначити, що системи скремблювання піддаються аналізу, оскільки не враховують інформаційні характеристики повідомлення і зашифрований сигнал зберігає дублювання вихідного сигналу, що особливо важливо для голосових повідомлень.

Застосування методів скремблювання для захисту голосових сигналів в цифровому вигляді не знайшло широкого застосування, так як використовувані для захисту цифрові перетворення сигналів досить трудомісткі, вимагають апаратної підтримки (сигнальних процесорів) для ефективної реалізації, характеризуються високою залишковою розбірливістю і є системами тимчасового опору.

Перераховані вище недоліки систем скремблінгу, з одного боку, і стрімкий розвиток цифрових систем зв'язку, з іншого, призвели до того, що в даний час пріоритет в області захисту голосових повідомлень зміщується в бік дискретних систем захисту повідомлень.

1.2. Способи захисту цифрових повідомлень.

Цифрове голосове повідомлення - це послідовність виміряних значень (вибірок) аналогового сигналу через рівні проміжки часу. Кожен зразок має

кінцеве число дискретних рівнів, представлених в числовій формі, тому до цифрових сигналів застосовні методи шифрування. Шифрування — це параметризована процедура, яка використовує оборотне перетворення, набір функцій, які ін'єктивно відображають набір відкритих текстів (вхідних даних) у набір шифротекстів (виходів). Параметр процедури шифрування - це ключ, який вибирає перетворення з набору функцій. Можна побудувати формальну алгебраїчну модель шифру [5].

Нехай *X, K, Y* — скінченні множини можливих відкритих текстів, ключів і шифротекстів відповідно; *E* - сукупність всіх можливих правил шифрування даного класу; *D* - це набір усіх можливих правил дешифрування даного класу; відображення $E_k: X \longrightarrow Y$ – правило шифрування ключа $k \in K$, що позначається як $E_k(X)$, відображення $D_k: Y \to X$ – правило розшифрування ключа $k \in K$, що позначається як $D_k(Y)$

Тоді шифр (система шифрів, секретна система) [6] - це множина введених множин $\Sigma_A = (X,K,Y,E,D)$, для яких виконуються наступні властивості:

1. $\forall x \in X$, $k \in K$ виконується рівняння $D_k(E_k(x)) = x$;
2. $Y = \bigcup_{k \in K} E_k(X)$.

$\sum_A$ – алгебраїчна модель шифру, яка відображає основні властивості дійсних шифрів. Умова 1 означає вимогу однозначного розшифрування, а умова 2 означає, що будь-який елемент $y \in Y$ може бути представлений у вигляді $Ek(x)$ для відповідних елементів **x** X і $k \in K$.

У загальному вигляді схема системи довільних шифрів представлена на рис.1.1:

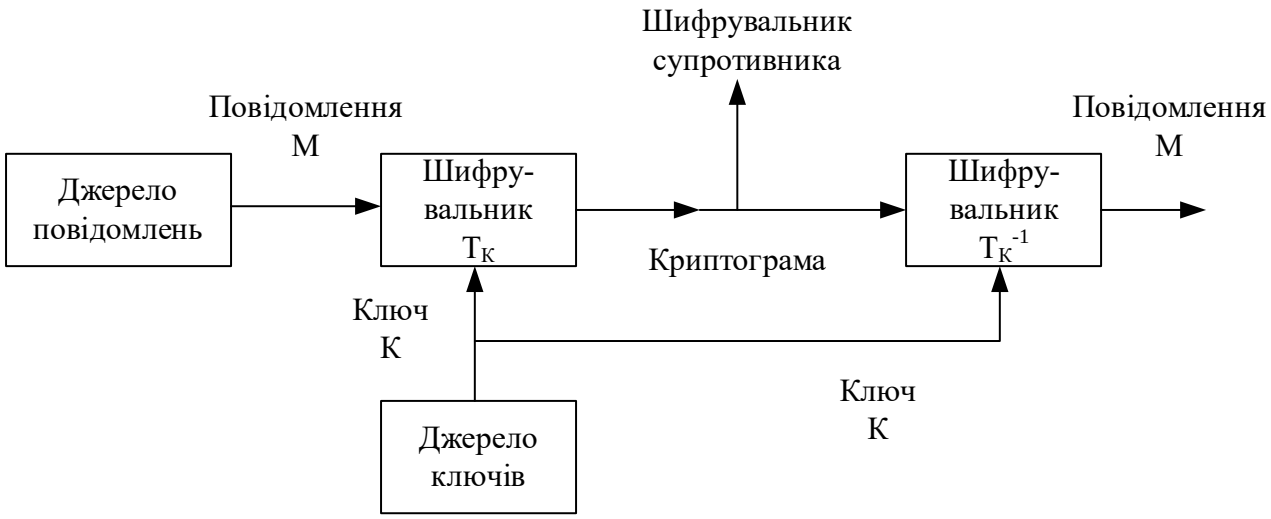


Рисунок 1.1 – Схема системи узагальнених шифрів.

В даний час існують шифри з секретним ключем (private key, symmetric systems) - для операцій шифрування і дешифрування використовується один і той же ключ і системи з відкритим ключем (асиметричні криптосистеми), для яких ключі шифрування і дешифрування різні [7, 8]. В рамках даної роботи були розглянуті симетричні криптографічні системи з секретним ключем, так як асиметричні системи з відкритим ключем характеризуються набагато меншою швидкістю роботи і високою трудомісткістю генерації ключів, що обмежує їх використання для вирішення проблеми захисту голосових повідомлень в цифровому вигляді.

Існує два типи симетричних алгоритмів: блокові шифри і потокові шифри. Блокові шифри працюють на блоках відкритого тексту і, при використанні одного і того ж ключа шифрування, перетворюють одні і ті ж блоки відкритого тексту в ідентичні блоки шифротексту. Потокові шифри перетворюють потік елементів відкритого тексту (біти, байти, n-байтові слова) в потік шифротексту, і кожен раз, коли одні й ті ж елементи відкритого тексту шифруються, вони створюють різні елементи шифротексту. Потокові шифри являють собою модель одноразового шифроблока і характеризуються простотою реалізації, так як в них здійснюється гамування відкритого тексту з виходом генератора псевдовипадкових послідовностей (PSP), але труднощі в побудові потокових

шифрів пов'язані з побудовою стійкого генератора PSP. Блокові шифри є складними нелінійними підстановками і характеризуються високою обчислювальною складністю, а також можуть використовуватися як вбудований шифротекст (але зі значно меншою швидкістю) в режимах зворотного зв'язку шифру (CFB) і зворотного зв'язку на виході (OFB) [9].

Аналіз більшості відкритих блочних шифрів показує, що сучасний блоковий шифр розбиває вхідний відкритий текст на блоки даних фіксованої довжини, кожен з яких перетворюється окремо і незалежно від інших блоків, як правило, за власним ключем. Більшість блокових криптографічних алгоритмів побудовані за ітераційним принципом і являють собою послідовність повторюваних перетворень (раундів) над блоком відкритого тексту з різними параметрами (ключами).

Відкритий текст розбивається на блоки даних (при необхідності останній блок доповнюється розміром блоку), кожен блок проходить кілька раундів шифрування, після чого блоки об'єднуються і формується криптограма.

Раунд (функція раунду, основний етап криптографічного перетворення) криптографічного перетворення - це послідовність криптографічних операцій, що застосовуються до блоку даних. Послідовне використання раундової функції з різними входами і параметрами необхідно для збільшення обчислювальної потужності алгоритму, тому кожен шифр містить кілька таких раундів, а шифр взагалі може складатися з різних раундових функцій. У сучасній криптографії кількість раундів коливається від 8 до 32, для зручності виконання раунди зазвичай ідентичні.

Для побудови стійких раундових функцій використовується комбінаційний підхід до побудови шифрів, запропонований в [9], який може бути описаний за допомогою математичного апарату алгебри секретних систем [9], що дозволяє абстрагуватися від конкретних характеристик використовуваного перетворення (секретної системи). Нехай секретна система T буде визначена як сімейство однозначно обернених відображень E_i множини

можливих повідомлень X в множині криптограм Y, а відображення E_i має ймовірність p_i.

Комбінації декількох секретних систем можуть бути використані для отримання нової секретної системи. Якщо системи, які потрібно об'єднати, мають однаковий простір повідомлень, можна визначити зважену суму:

$$S = p_1T + p_2R + ... + p_mU,$$

де $T,R,...,U$ – секретні системи, $p_1,p_2,...,p_m$ – ймовірність використання системи $T,R,...,U$ відповідно, причому $\sum_{i=1}^{m} p_i = 1$.

Другий спосіб комбінування секретних систем - це добуток, коли область визначення (мовний простір) однієї системи ототожнюється з областю значень (криптограмовим простором) іншої системи:

$$S = T \cdot P \cdot ... \ U.$$

Результуюча операція S є послідовним застосуванням систем $T,R,...,U$ до нашої мови, а ключ S складається з ключів T,R,...,U в припущенні, що ключі вибираються незалежно і відповідно до їх початкових ймовірностей. Таким чином, якщо m ключів системи T вибрані з імовірностями $p_1,p_2,...p_m$, , а n ключів системи R обрані з імовірностями $q_1,q_2,...q_n$, то система $S = RT$ має не більше $m \bullet n$ ключів з імовірностями $p_i \cdot q_j$. Загалом, операція добутку шифрів є некомутативною, але в окремих випадках має місце комутативність.

Тоді раунд блокового шифру можна представити у вигляді добутку деяких елементарних перетворень:

$$Round = \prod_{i=1}^{m} T_i\left(x, k_i\right),$$

$T_i(x, k_i)$ – де є результатом використання *i*-ої секретної системи для перетворення блока даних *x* на ключі k_i системи.

Раундовий ключ K_i в цьому випадку є добутком *m* ключів елементарних перетворень, тому можна стверджувати, що раундовий ключ є ключовим вектором розмірності *m*.

Тоді блоковий шифр, що складається з *n* ідентичних раундів, може бути визначений як

$$Cipher = (Round)^n = \left(\prod_{i=1}^{m} T_i(x, k_i) \right)^n .$$

Ключ K блокового шифру, що складається з *n* раундів по m перетворень кожен, як правило, є ключовим вектором розмірності $m \cdot n$ (для різних раундів). Стійкість системи заснована на секретності ключа $k \in K$.

Для побудови раундових функцій симетричних блочних шифрів використовуються комбінації трьох основних операцій: перестановок, підстановок (замін) і гамування. Кількість, послідовність і параметри цих операцій визначають основні якості шифру (стійкість, простота реалізації і швидкість виконання алгоритму).

Перестановки

Перестановкою σ на множині Z_N називається гомоморфізмом Z_N, при якому букви вихідного тексту переставляються в залежності від їх порядкового номера в повідомленні, тобто вихідний текст переставляється в інший порядок. Для того, щоб показати, що символ повідомлення був переміщений з позиції *i* в позицію σ(*i*), де $0 \le i \le N-1$, позначення

$$\sigma = (\sigma(0), \sigma(1), ..., \sigma(N-1)).$$

Перестановка σ встановлює взаємооднозначне відображення множини Z_N, що складається з N елементів, на себе:

$$\sigma : s_i \to s_{\sigma(i)}, 0 \le i \le N-1$$

Кількість перестановок з множини $Z_N(0,1,\ldots,N-1)$ дорівнює $N!$

Таким чином, використання перестановки не виводить за межі вихідного повідомлення (в сенсі використовуваних символів алфавіту), а лише змінює порядок символів відкритого тексту в криптограмі.

Операція перестановки широко використовується в скремблюванні, коли вихідний сигнал ділиться на фрагменти, а кожен фрагмент, в свою чергу, ділиться на сегменти, після чого відрізки переставляються всередині фрагмента і фрагменти переставляються. У симетричних блочних шифрах перестановка використовується для вирівнювання статистичних характеристик відкритого тексту. Перестановки використовуються в якості вхідних та вихідних таблиць перестановки (і задаються за допомогою таблиць перестановок) або як зміщення символів відкритого тексту (задаються аналітично величиною та напрямком перестановки). Шифри перестановки нестійкі проти частотного аналізу та аналізу на основі вибраного відкритого тексту.

Підстановки

Підстановкою π на алфавіті Z_M є автоморфізм Z_M, при якому букви вихідного тексту x замінюються буквами шифротексту $\pi(x)$:

$$Z_m \to Z_m; \pi : x \to \pi(x).$$

Множина всіх перестановок називається симетричною групою Z_M - $SYM(Z_m)$, що є групою з дією добутку, тобто операцією, що володіє такими властивостями: замкнутість, асоціативність, наявність нейтральних і обернених

елементів. Число можливих підстановок в $SYM(Z_m)$ називається порядком і дорівнює $M!$.

Ключем підстановки k для Z_M є послідовність елементів симетричної групи Z_M:

$$k = (p_0, p_1, ..., p_{n-1}, ...), \ \ p_n \in SYM(Z_m), 0 \le n \le \infty.$$

Підстановка, що визначається ключем k, є криптографічним перетворенням Ek, за допомогою якого здійснюється перетворення n-грами вихідного тексту $(x_0, x_1, \ldots, x_{n-1})$ у n-граму шифротексту:

$$(y_0, y_1, \ldots, y_{n-1})$$
$$y_i = p(x_i), \ 0 \le i \le n,$$

де n – довільне ціле число (n = 1,2,..). Буква шифротексту y_i є функцією лише i-*го* компонента ключа p_i та i-ої букви вихідного тексту x_i. E_k називається підстановкою один до одного, якщо p є незмінним для будь-якого i, інакше E_k називається багатозначною підстановкою. Багатозначна підстановка визначається ключем $\pi = (\pi_1, \pi_2, ...)$, що містить принаймні дві різні заміни. Якщо алфавіти відкритого тексту і криптограми збігаються, то підстановка називається моноалфавітною, інакше вона називається мультиалфавітною.

Підстановка визначається таблицею підстановки, що містить пари відповідних символів "вихідний текст - шифротекст". Загалом, підстановка є операцією кодування, оскільки, як і операція кодування, вона переводить символи з одного алфавіту на інший. У криптографії ці алфавіти часто збігаються, тобто відбувається заміна символів оригінального тексту на інші (того ж алфавіту) за більш-менш складним правилом. Різниця між підстановкою і перестановкою полягає в тому, що таблиця підстановки (ключ підстановки) вказується для всього алфавіту, а для перестановки вона вказується тільки для

повідомлення, тобто підстановка, в загальному випадку, виводить за межі вихідного повідомлення в сенсі використовуваних символів, а перестановка - ні.

Будь-який шифр може бути представлений у вигляді підстановки (або суперпозиції підстановок), в якій роль ключа підстановки виконує складна комбінація ключа шифру разом з порядком використовуваних елементарних операцій. Практично всі сучасні шифри можуть бути представлені підстановкою, але повне визначення шифрів у вигляді таблиці підстановки в даний час не використовується, так як обсяг таблиці стає дуже великим, що значно ускладнює аналіз. Замість цього використовується аналітичне завдання шифру за більш-менш складним законом.

Таблиці підстановки в сучасних блокових шифрах використовуються як вузли нелінійності (підстановки) для вирівнювання статистичної надмірності вхідного відкритого тексту і багато в чому визначають стійкість шифру [3]. Підстановки також доступні для криптоаналітичних досліджень, оскільки їх стійкість зменшується зі зменшенням довжини ключа, а збільшення довжини ключа та складності алгоритму збільшує складність реалізації.

Гамування

Гамування є одним з найбільш широко використовуваних криптографічних перетворень. Принцип гамма-шифрування полягає в накладенні псевдовипадкової послідовності символів (шифр гамма) на відкриті дані оборотним способом (шляхом додавання цифр або віднімання за модулем) таким чином, що зчитування інформації було б неможливим без знання шифрованої гами. Процес розшифровки даних зводиться до повторного застосування гами шифру до зашифрованих даних. Операція, що використовується для накладення гами, повинна бути однозначно розв'язною щодо відкритого тексту, за умови, що відомий зашифрований текст і ключ. В даний час для цієї мети використовується операція "додавання за модулем 2" (виключне OR, XOR), так як вона вимагає найменш складної логіки з усіх можливих операцій, обернена сама по собі, і ефективно реалізована на всіх платформах.

Отримана криптограма не може бути виявлена, якщо гамма шифру випадкова. Якщо гамма-період перевищує довжину всього шифротексту і жодна частина вихідного тексту не відома, то шифр може бути вирішений тільки повним пошуком всіх можливих ключів, і цей шифр є досконалим шифром, для якого строго доведено відсутність більш ефективного алгоритму злому, окрім метода повного перебору [3,6].

Для побудови шкали необхідно використовувати випадкові послідовності, але на практиці використовуються детерміновані генератори псевдовипадкових послідовностей (ПСП), вихідна послідовність яких близька до випадкової за своїми властивостями, але має кінцевий (хоча і дуже довгий) період. Перевагою генераторів PSP є можливість отримання ключів будь-якого розміру з невеликою кількістю секретної інформації (початкових значень). Найбільш поширеними є конгруентні генератори PSP, регістри лінійного зсуву (LRS) та їх варіації, а також нелінійні генератори.

1.3. Вплив ентропії та надмірності на стійкість шифрів

Для застосування критеріїв оцінки стійкості шифрів використовується імовірнісне визначення шифру [6].

Повідомлення, відправлене до того, як воно дійде до одержувача, є невизначеним і для нього, і для криптоаналітика. При відправці повідомлень X_1, $X_2,...,X_n$ з ймовірністю $p_1,p_2,...,p_n$ відповідно, невизначеність (ентропія) повідомлення служить мірою невизначеності повідомлення [6]:

$$H(X) = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i.$$

Фізичний сенс цієї величини – це кількість бітів інформації, які необхідно передати, щоб повністю виключити невизначеність повідомлення. Відсутність будь-якої апріорної інформації про повідомлення X, крім її розміру в N бітах,

означає, що всі можливі 2^N варіанти еквівалентні, і тоді невизначеність повідомлення дорівнює його розміру:

$$H(X) = -2^N \cdot 2^{-N} \cdot \log_2 2^{-N} = N = |X|,$$

де $|X|$ – розмір повідомлення X у бітах.

Коли шифротекст перехоплюється, це значення стає апостеріорною умовною невизначеністю:

$$H(X) = -\sum_{i=1}^{n} p(X_i / Y) \cdot \log_2 p(X_i / Y),$$

де $p(X_i/Y)$ – ймовірність того, що вихідне повідомлення дорівнює X_i за умови, що результатом його кодування є Y. Кількість інформації про вихідний текст, яку можна витягнути з перехопленого шифротексту [3]:

$$I = H(X) – H(X/Y).$$

Ця величина характеризує зменшення невизначеності відкритого тексту при отриманні відповідного зашифрованого тексту в порівнянні з його попередньою невизначеністю. I = 0 означає, що знання шифротексту не дозволяє зменшити невизначеність відповідного відкритого тексту, поліпшити його оцінку і підвищити ймовірність його правильного визначення. Для таких шифрів неможливо збільшити ймовірність правильної розшифровки через відсутність альтернатив у криптоаналітика у виборі результату розшифровки [12]. Ще одним часто використовуваним типом шифрів є досконалі шифри, для яких невизначеність алгоритму шифрування $H(E) = \infty$, а це означає, що відкрити його ефективніше інакше як методом повного неможливо [9]. При цьому досконалі системи завжди досконалі, а ідеальні системи не завжди ідеальні - досконаліть

шифру не ґрунтується тільки на рівномірності розподілу відкритих даних [3]. Однак на практиці важко підтримувати повну невизначеність щодо шифру. Тому алгоритм повинен бути перенастроюваним таким чином, щоб у разі компрометації шифру не було необхідності повністю замінювати алгоритм на всіх вузлах шифрування, де він використовується; у ньому повинна бути присутня легко замінна частина (ключові параметри). Прикладом ідеального шифру є одноразова гамма Вернама - накладення на відкриті дані T ключа K однакового розміру, складеного зі статистично незалежних бітів, які приймають можливі значення з однаковою ймовірністю:

$$Y` = X \oplus K.$$

Детальний аналіз можливості розшифровки повідомлення, зашифрованого недосконалим шифром, дається в [14] і пов'язаний з ентропією мови і надмірністю відкритого тексту. Ентропія мови Λ (з числом букв n) характеризує міру середньої кількості інформації на букву тексту мови Λ, і обчислюється як

$$H_\Lambda = -\sum_{i=1}^{n} p_i \log_2 p_i.$$

Для реальних мов людського спілкування, які мають статистичні залежності, для отримання більш точного значення H_Λ необхідно використовувати моделі більш високого рівня (для аналізу ймовірностей появи біграм, триграм, k-грам). Для статистично незалежних алфавітів ентропія на букву буде виражатися як

$$H_\Lambda = \log_2 n.$$

Надмірність R_Λ мови Λ показує, яку частину букв можна опустити без втрати змісту, і виглядає так:

$$R_\Lambda = \frac{\log_2 n - H_\Lambda}{\log_2 n} = 1 - \frac{H_\Lambda}{\log_2 n}.$$

Дослідження [9] показують, що з підвищенням рівня моделі для реальних мов спостерігається помітне зниження ентропії і збільшення надмірності. Поняття надмірності тісно пов'язане з дистанцією унікальності - кількості шифротексту, необхідного для одиничного рішення криптограми. Якщо об’єм шифротексту менше відстані, то існує декілька варіантів відкритого тексту. Деякий «випадковий шифр» в принципі піддається розшифруванню при необмежених ресурсах криптоаналітика, якщо обсяг перехопленого шифротексту перевищує відстань унікальності цього шифру L:

$$L = \frac{\log_2 |K|}{R_\Lambda \cdot \log_2 n},$$

де |K| – потужність набору ключів (кількість можливих ключів).

Однак, крім збільшення довжини ключа, одним із способів збільшення стійкості шифру є усунення надмірності відкритого тексту. Таким чином, при $R_\Lambda \to 0$ це, очевидно, $L \to \infty$. Відсутність надмірності у вихідному тексті означає, що розшифровка на будь-якому ключі дає «правильні» вихідні дані, і немає причин віддавати перевагу одному варіанту перед іншим. Умова L = оо відповідає поняттю ідеального шифру, що завжди залишає невизначеність для криптоаналітика у виборі результату криптоаналізу. Однак ідеальність шифру не означає його досконалість, так як критерієм досконалості є відсутність більш ефективних методів злому, ніж повне перерахування, а для ідеального шифру – відсутність однозначної розшифровки. Досконалі системи завжди ідеальні, але не всі ідеальні системи ідеальні. Тому одним з підходів до побудови надійних шифрів є наближення властивостей розробленого алгоритму до ідеальних і досконалих шифрів.

Різні методи стиснення інформації використовуються як спосіб зменшення надмірності даних. Хоча пріоритетною задачею застосування алгоритмів стиснення є зменшення обсягу, це досягається оптимальним кодуванням, в основі якого лежить принцип ентропійного кодування елементів, що часто зустрічаються короткими кодами, і рідкісних довгими кодами. У застосуванні до криптографії це означає бажаність стиснення при використанні шифрів, що зменшує кількість відкритого тексту (і, відповідно, збільшує швидкість шифрування) і зменшує надмірність даних, тим самим ускладнюючи криптоаналіз.

1.4. Критерії оцінювання криптографічних алгоритмів

У криптографії криптографічна стійкість традиційно визначається як стійкість до спроб розшифрування повідомлень, тобто кількісна міра того, наскільки легко криптоаналітик може зламати шифр, що вимірюється мінімальною кількістю операцій шифрування/дешифрування, необхідних для визначення ключа (або відкритого тексту) у найкращому випадку атаки. Основне правило криптографії по відношенню до питання криптографічної стійкості полягає в наступному: злом шифру з метою зчитування секретної інформації повинен коштувати набагато дорожче (вартість виражається в деяких ресурсах - кількості чіпів для реалізації атаки, обсязі пам'яті і т.д.), ніж реальна вартість інформації.

Поняття криптографічної стійкості тісно пов'язане зі складністю криптоаналітичної атаки на алгоритм шифрування і характеризується трьома значеннями [3]:

Складність даних – обсяг перехоплених даних, необхідний для успішної криптоаналітичної атаки на алгоритм шифрування.

Обчислювальна складність – це кількість операцій, необхідних для успішної атаки на алгоритм шифрування.

Складність пам'яті – обсяг пам'яті, необхідний для успішної атаки.

Для оцінки стійкості можна взяти максимальне з цих значень або скористатися оцінкою, заснованою на узагальненому критерії.

За криптографічною стійкістю всі шифри поділяються на досконалі (які не мають більш ефективних алгоритмів відкриття, крім методу повного перебору) і недосконалі (не мають абсолютної стійкості), стійкість яких ґрунтується на обчислювальній складності вирішення певної математичної задачі. Обчислювальна потужність оцінюється за допомогою кількості операцій (w) певного типу, необхідних для розшифрування повідомлення або визначення ключа. Найзручнішою операцією для отримання показників стійкості є цикл перевірки одного ключа. Трудомісткість розшифровки залежить від характеру і кількості інформації, яка є в розпорядженні аналітика:

1. Аналіз на основі тільки шифротексту – аналітик знає механізм шифрування і йому доступний тільки шифротекст розміру n, який відповідає моделі зовнішнього зловмисника, який має фізичний доступ до лінії зв'язку.

2. Аналіз на основі заданого відкритого тексту – криптоаналітик знає зашифрований текст і певну частину вихідної інформації, а в окремих випадках відповідність між шифротекстом і вихідним текстом, тобто аналітик має зашифроване повідомлення розміру n і відповідний відкритий текст. Така атака можлива при шифруванні стандартних документів, підготовлених за стандартними формами, коли певні блоки даних повторюються і відомі.

3. Аналіз на основі довільно обраного відкритого тексту - криптоаналітик може ввести спеціально виділений текст в шифрувальний пристрій і отримати криптограму, сформовану під контролем секретного ключа, тобто аналітик має в своєму розпорядженні результат шифрування для довільно обраного масиву відкритих даних розміру *n,* що відповідає моделі внутрішнього зловмисника, коли в атаці на шифр беруть участь особи, які не знають секретного ключа, але відповідно до своїх службових повноважень мають доступ до шифрувального пристрою;

4. Аналіз на основі довільно підібраного шифротексту - противник має можливість підставляти для розшифровки вигадані шифротексти, які

підбираються особливим чином, щоб з текстів, що надійшли на виході дешифратора, він міг обчислити ключ шифрування з мінімальними зусиллями, тобто аналітик має в своєму розпорядженні результат розшифровки довільно обраного зашифрованого повідомлення розміру n, що відповідає доступу до пристрою розшифровки;

5. Адаптивна текстова атака – зловмисник багаторазово підставляє тексти для шифрування (або розшифровки), і кожен новий шматок даних вибирається в залежності від отриманого результату перетворення попередньої порції.

Середній обсяг роботи W(N), необхідний для визначення ключа з криптограми, що складається з N букв, виміряних в елементарних операціях, називається робочою характеристикою шифру. Ця величина приймається як середнє для всіх ключів і всіх повідомлень з відповідними їм ймовірностями.

Нижні межі трудомісткості досягаються при деяких кінцевих значеннях параметра N, тому що при необмеженому його збільшенні трудомісткість аналізу буде прагнути до певної межі, званої досягнутою оцінкою виробничої характеристики. Як зазначалося в [7], для недосконалих шифрів немає можливості отримати точне значення трудомісткості аналізу. Всі оцінки грунтуються на перевірках стійкості шифрів до відомих на даний момент видів криптоаналізу, і немає ніякої гарантії, що в найближчому або більш віддаленому майбутньому не будуть розроблені нові методи аналізу, які значно знизять її. Тому стійкість недосконалих шифрів емпірично обґрунтовується як стійкість до відомих на сьогоднішній день видів криптоаналізу.

В даний час в криптоаналізі виділяють такі основні методи:

1. Повний пошук можливих ключів;
2. Частотний аналіз;
3. Диференціальний метод;
4. Лінійний метод.

Повний пошук можливих ключів

У криптоаналізі з відомим вихідним кодом, який є найбільш релевантним, використовується єдиний опублікований метод - це повний перебір ключів.

Основним недоліком цього методу є експоненціальна залежність часу криптоаналізу від довжини ключа, тому через високу трудомісткість цей метод вибирають в якості верхньої межі при оцінці криптографічної стійкості шифру - якщо задачу злому шифру можна вирішити тільки методом повного перебору, реалізація якого буде неприйнятною з фінансових або часових міркувань, то цей шифр вважається стійким. Цей метод використовується і для оцінки ефективності інших алгоритмів криптоаналізу – всі інші методи повинні забезпечувати значно меншу трудомісткість.

Існують оптимізації цього методу, пов'язані з використанням атак за словником. Ідея полягає в тому, що відправник повідомлення віддасть перевагу використанню легкої для запам'ятовування послідовності ключів, а не повністю випадкової. Тому, перш за все, необхідно випробувати найбільш ймовірні ключі (і їх комбінації) зі словника, які дозволяють істотно знизити трудомісткість злому шифру. Практичні дослідження показали високу ефективність такого підходу.

Частотний аналіз

Частотний аналіз враховує частотні характеристики текстів на реальних мовах, які мають високу надмірність і наявність стабільних частот комбінацій символів (k-грам і словоформ мови). Принцип методу полягає в підрахунку частот символів в криптограмі і порівнянні їх з розрахунковими частотами для моделі відкритого тексту. Перевага методу в тому, що його можна використовувати для шифрів з різними алфавітами, для відкритого тексту і криптограм. Для побудови математичної моделі відкритого тексту використовується побудова частотних характеристик відкритого тексту, розрахованих з необхідною точністю при вивченні текстів достатньої довжини.

Відкритий текст розглядається як реалізація стаціонарного ергодичного стохастичного процесу з дискретним часом і кінцевим числом станів, а модель являє собою однорідний ланцюг Маркова [39].

Більш високий порядок наближення відповідає більш зв'язному тексту, який виходить з виводу моделі. Критерії розпізнавання ґрунтуються або на

стандартних методах розрізнення статистичних гіпотез, або на основі заборонених k-грамів (деякі рідкісні k-грами оголошуються забороненими і їх наявність в отриманому тексті означає отримання «безглуздої» послідовності символів). При використанні спеціальних алфавітів (для нетекстової інформації) необхідно вибудовувати нові формалізовані критерії відкритого тексту з урахуванням специфіки формування вхідної послідовності, що саме по собі є складним завданням, пов'язаним з додатковим дослідженням.

Побудова критеріїв для мови, що характеризується відсутністю статистичних залежностей (рівномірним розподілом ймовірностей появи символів), є майже нерозв'язною задачею.

Диференціальний метод криптоаналізу

Диференціальний криптоаналіз (DCA) — це спроба зламати секретний ключ блокових шифрів, які засновані на повторному застосуванні криптографічно слабкої операції цифрового шифрування (функція раунду) *r* разів. Аналіз передбачає, що кожен цикл використовує різний підключ шифрування. CSA може використовувати як вибрані, так і відомі відкриті тексти. Успіх таких спроб зламати циклічний шифр залежить від існування диференціалів *(r-1)*-го цикли, які мають високу ймовірність. Суть методу полягає в знаходженні диференціалів для останнього циклу, на підставі яких можна визначити підключ циклу, після чого процедура повторюється.

Відмінною особливістю диференціального аналізу є те, що в ньому практично не використовуються алгебраїчні властивості шифру (лінійність, афінність, транзистивність, замкнутість і т.д.), але базується лише на нерівномірному розподілі ймовірності диференціалів.

Лінійний метод криптоаналізу.

Метод передбачає, що криптоаналітик знає відкритий текст і відповідні шифротексти. Як правило, шифрування використовує додавання тексту ключа за модулем 2 та операції розсіювання та перемішування. Метод заснований на знаходженні найкращої лінійної апроксимації (після всіх циклів шифрування) для виразу, що описує вихід шифру, а потім для знаходження кожного біта

самого ключа необхідно вирішити систему лінійних рівнянь для відомих лінійних комбінацій цих бітів, але ця процедура не є складною, так як складність рішення системи лінійних рівнянь описується многочленом не більше третини ступеня довжини ключа.

Автоматизація методів криптоаналізу перебору ключів з використанням комп'ютерів вимагає створення адекватної математичної моделі відкритого тексту для оцінки отриманих результатів. Багато в чому успіх криптоаналізу ґрунтується на надмірності відкритих текстів на реальних мовах [6]. Тому перед шифруванням бажано піддати дані перетворенню, що зменшує надмірність, яким може бути стиснення інформації. Переваги використання стиснення над шифруванням, крім зменшення надмірності, можуть прискорити трудомісткий процес шифрування за рахунок зменшення обсягу відкритих даних.

1.5. Ітераційні блокові шифри

Більшість сучасних блокових шифрів (додаток 4) мають схожу архітектуру, що обумовлено вимогами до характеристик практичних шифрів. Практичний блоковий шифр повинен бути технічно застосовним для шифрування масивів даних довільної довжини, і повинен бути реалізований у вигляді пристрою (або програми) з обмеженим обсягом пам'яті. Поєднання обох вимог неминуче призводить до алгоритму, в якому шифрування виконується крок за кроком, а один блок шифрується за один крок. Розмір блоку майже завжди приймається постійним, за винятком останнього блоку даних, який при необхідності доповнюється до розміру блоку. З міркувань стійкості розмір блоку не повинен перевищувати розмір ключа, тому вибір розміру блоку має сильний вплив на стійкість шифру. Якщо розмір блоку невеликий, то блоковий шифр не може бути стійким, так як при відносно невеликому обсязі відомого вихідного тексту найбільш поширені значення вхідних блоків будуть вичерпані. В даний час використовуються шифри з розміром входу 64-256 біт. Чим більше розмір блоку шифрування, тим більш висока стійкість досяжна, так як більший розмір

блоку вимагає розгляду більшої кількості взаємодіючих бітів і більш різноманітних можливих алгоритмів перетворення, але занадто великий розмір блоку вимагає багато ресурсів як для апаратної, так і для програмної реалізації.

Були сформульовані необхідні умови стійкого шифру - змішування і розсіювання. Розсіювання - це властивість шифру, при якому один біт вихідного тексту впливає на кілька бітів шифротексту, оптимально на всі символи в межах одного блоку. Якщо ця умова виконується, то при шифруванні двох блоків даних з мінімальними відмінностями між ними повинні бути отримані найбільш різнорідні блоки шифротексту. Точно така ж схема повинна бути вірна і для залежності шифротексту від ключа - один біт ключа повинен впливати на кілька бітів шифротексту. Як зазначено в [7], найкращим способом реалізації такого «розсіювача» є операція перестановки, яка в сучасних шифрах реалізована у вигляді Р-блоків.

Перемішування - це властивість шифру приховувати залежності між символами вихідного тексту і зашифрованого тексту. Якщо шифр досить добре змішує біти вихідного тексту, то відповідний зашифрований текст не містить жодних статистичних закономірностей для стороннього спостерігача.

Якщо шифр в достатній мірі володіє властивостями змішування і розсіювання (забезпечує достатній рівень дифузії), то будь-які зміни в блоці відкритих даних призводять до того, що, з точки зору спостерігача, всі символи в зашифрованому блоці будуть отримувати нові значення, однаково ймовірні в області їх визначення і незалежні один від одного.

Важливим припущенням при проектуванні шифру є правило Керкгоффса: в криптоаналізі система шифрування вважається відомою, а стійкість визначається тільки секретністю ключа шифрування. Тому, з одного боку, бажано мати якомога менший ключ (який в ідеалі повинен бути запам'ятаний), а з іншого - необхідно забезпечити високий рівень стійкості алгоритму. Тому проблема криптографії в даний час полягає в побудові шифрів з високою стійкістю і якомога меншим ключем. Для оцінки рівня стійкості часто

використовуються економічні міркування. Таблиця 1.1 показує приблизні часові та кошторисні витрати (для можливих опонентів) для різних ключових довжин.

Таблиця 1.1 – Оцінка середнього часу відкриття апаратних засобів за допомогою повний перебору

Вартість, млн. дол.	Довжина ключа, біти					
	40	56	64	80	112	128
0.1	2 сек	35 год.	1 рік	70000 років	10^{14} років	10^{19} років
1	0,2 с	3,5 години	37 днів	7 000 років	10^{13} років	10^{18}років
10	0,02 сек	21 хв	4 дні	700 років	10^{12} років	10^{17}років
100	2 мс.	2 хв	9 ранку.	70 років	10^{11} років	10^{16} років
1000	0,2 мс	13 сек.	1 година.	7 років	10^{10} років	10^{15} років
10 000	0,02 мс	1 сек	5.4 хв.	245 днів	10^{9} років	10^{14} років
100 000	2 мксек.	0,1 с	32 сек	24 дні	10^{8} років	10^{13} років
1 000 000	0,2 мксек	0,01 сек	3 сек	2,4 дні	10^{7} років	10^{12} років
10 000 000	0,02 мксек	1 мс.	0,3 сек	6 ранку.	10^{6} років	10^{11} років

Вартість характеризує можливість побудови спеціалізованого апаратного пристрою для злому шифру. Як видно з таблиці 1.1 Найбільш безпечною довжиною ключа на даний момент, навіть з урахуванням розвитку комп'ютерних технологій, є довжина ключа в 128 біт, що становить 16 байт. Більшість блокових шифрів належить до популярної архітектури мереж Файстеля, фундамент якої був закладений на початку 70-х років 20 століття.

Шифр з використанням такої конструкції гарантовано буде оборотним, якщо є можливість відновити вихідні дані f в кожному раунді, а сама f не обов'язково оборотний. Недоліками такої архітектури є велика кількість раундів для забезпечення прийнятної дифузії і стійкості.

Згідно з термінологією, введеною в 1.2, раунд блокового шифру є продуктом деяких елементарних перетворень:

$$Round = \prod_{i=1}^{m} T_i\left(x, k_i\right),$$

де $T_i\left(x, k_i\right)$ – результат використання *i*-ї секретної системи для перетворення блоку даних *x* на ключ k_i системи.

Позначимо розглянуті в попередньому підрозділі операції перестановки, підстановки і гами як секретні системи *P* (перестановка), *S* (підстановка) і *G* (гамма), а сам факт використання ключів *pk, sk, gk* як $\boldsymbol{P_{pk}}$, $\boldsymbol{S_{sk}}$ і $\boldsymbol{G_{gk}}$ відповідно.

Тоді раунд криптографічного перетворення можна представити у вигляді добутку цих систем:

$$Round = P_{pk} \cdot S_{sk} \cdot G_{gk}$$

Очевидно, що круглим ключем для такого перетворення відповідно до введеної алгебри секретних систем є $K_r = pk \cdot sk \cdot gk$.

Як було показано в попередньому розділі при розгляді елементарних криптографічних операцій, кожне з перетворень має свій секретний ключ, який забезпечує секретність інформації, що захищається. Однак в даний час спостерігається досить парадоксальна ситуація, пов'язана з тим, що в якості ключа для алгоритмів блочного шифрування використовується тільки гамма шифр (в розглянутому прикладі - ключ gk), накладаються на відкриті дані. Мабуть, це пов'язано з тим, що більшість алгоритмів блочного шифрування побудовані поверх одного з перших відкритих алгоритмів - DES, який був орієнтований в основному на апаратну реалізацію. У зв'язку з особливостями апаратної реалізації того часу, використання змінних значень зсувів і значень S-блоків було дуже неефективним і в описі алгоритму ці дані наводилися у відкритому вигляді. Тому для більшості блочних алгоритмів стійкість визначається тільки секретністю гами, що призводить до необхідності розробки нових алгоритмів шифрування з більшою довжиною ключа. Тим часом, як зазначалося, використання додаткових параметрів раунду криптографічного

перетворення (таких як значення S-блоків) дозволяє значно збільшити довжину ключа алгоритму.

Тому можна стверджувати, що можливим підходом до підвищення стійкості алгоритмів (ефективної довжини ключа) є підхід, пов'язаний з параметризацією алгоритмів шифрування (введенням додаткових ключів для перестановок і підстановок). Такий підхід має безперечні переваги в програмній реалізації алгоритмів шифрування, оскільки дозволяє легко змінювати параметри таблиць підстановки і перестановки. Тому представляється актуальним розробка методів додаткової параметризації алгоритмів шифрування, орієнтованих на програмну реалізацію для збільшення ефективної довжини ключа. При виборі архітектури блокового алгоритму в даній роботі перевага віддається алгоритму Rijndael (варіант загальних мереж SP), що обумовлено наявністю наступних позитивних характеристик: байт-орієнтована архітектура, можливість ефективної реалізації легко замінюваних таблиць підстановки з перемішаним управлінням.

1.6. Постановка задачі на дослідження

Сучасні реалії такі, що необхідно розробляти методи захисту не тільки текстової інформації: масивів даних, голосових повідомлень і відеосигналів, що характеризуються великим обсягом займаної інформації. У зв'язку з розвитком цифрових систем зв'язку необхідно захищати мовну інформацію, представлену цифровими сигналами. Використання методів захисту аналогових голосових повідомлень для голосових повідомлень в цифровому вигляді недоцільно, так як виконувані перетворення на аналоговому сигналі в процесі програмної реалізації вимагають великої кількості трудомістких процедур або використання спеціалізованих сигнальних процесорів і не забезпечують достатньої стабільності.

Недоцільним також видається використання уніфікованих криптографічних алгоритмів, призначених для шифрування текстових

повідомлень з високим резервуванням і малими розмірами для захисту голосових повідомлень, оскільки вони не враховують специфіку формування вхідного потоку джерелом.

Беручи до уваги великий обсяг голосових повідомлень в цифровому вигляді, доцільно використовувати алгоритми цифрового стиснення голосу для збільшення швидкості процедур шифрування і зменшення надмірності.

Тип використовуваного алгоритму стиснення, як правило, залежить від характеристик сигналу, що передається. Для цифрового стиснення сигналу найбільш поширені алгоритми стиснення сигналу з втратами (ADPCM, дельта-перетворення (D-перетворення) і подібне), характеризується відновленням сигналу з певною (допустимою) похибкою при збереженні суб'єктивної повноти відновленої інформації. Одним з перспективних методів стиснення цифрових потоків з втратами є D-перетворення, яке характеризується простотою і високою швидкістю, особливо алгоритмів демодуляції. Алгоритми оптимізованих D-перетворень другого порядку, описані в [6, 12], мають досить високі якісні характеристики. У той же час ефективність цих алгоритмів можна підвищити, адаптувавши їх для кодування мовних сигналів. Висока продуктивність алгоритмів кодування і декодування на основі дельта-перетворення другого порядку відкриває перспективні можливості для програмної реалізації стиснення мовного сигналу в режимі реального часу. Тому представляє інтерес розгляд можливості використання алгоритмів D-перетворення для стиснення голосових повідомлень в цифровій формі і розробка алгоритмів шифрування, що враховують характеристики вхідного потоку відкритого тексту.

Розглянуті проблеми використання блочних шифрів обумовлюють необхідність досліджень в області розробки методів підвищення стійкості алгоритмів шифрування. Для підвищення стійкості розроблених алгоритмів пропонується використовувати параметризацію алгоритму шифрування, яка полягає у використанні складеного ключа з ключів всіх використовуваних криптографічних операцій, що дозволить збільшити швидкодію алгоритму за рахунок зменшення числа раундів шифрування при забезпеченні тієї ж

криптографічної стійкості. В силу цього можна сформулювати вимоги до розроблених алгоритмів:

1. Алгоритм повинен забезпечувати обробку даних великого обсягу і довільного розміру;

2. Алгоритм повинен допускати ефективну реалізацію програмного забезпечення;

3. Алгоритм повинен містити некомутативну комбінацію підстановок і перестановок;

4. Підстановки, що використовуються в алгоритмі, повинні визначатися як входними даними, так і ключем;

5. Довжина зашифрованого тексту повинна бути такою ж, як і довжина відкритого тексту;

6. Довжина ключа і відкритого текстового блоку повинна бути змінною, щоб відповідати різним вимогам безпеки.

1.7 Висновки

1. Аналіз методів захисту голосових повідомлень в аналоговій формі виявив серйозні недоліки методів скремблювання для захисту мовних повідомлень в цифровому вигляді (висока трудомісткість, висока залишкова розбірливість, мала довговічність), що дозволяє зробити висновок про недоцільність використання методів скремблювання для захисту цифрових мовних сигналів;

2. Огляд методів захисту дискретних повідомлень дозволяє сказати, що спеціалізованих методів захисту голосових сигналів в цифровому вигляді не існує з урахуванням специфіки їх формування джерелом;

3. У зв'язку з орієнтацією більшості існуючих блокових систем шифрування на використання єдиних алгоритмів захисту інформації актуальною є розробка спеціалізованого алгоритму захисту голосових повідомлень;

4. Аналіз основних операцій криптографічних перетворень (підстановки,

перестановки, гами) і композиційний симетричний блоковий шифр дозволяє стверджувати, що в блочних алгоритмах роль ключа виконує тільки гамма-константа, що зменшує ключовий простір алгоритму;

5. На основі аналізу проблем у сфері застосування існуючих криптографічних алгоритмів запропоновано використання параметризації (операцій перестановки та підстановки) для збільшення ефективної довжини ключа;

6. В результаті розгляду впливу статистичних характеристик (ентропії і надмірності мови) відкритого тексту на стійкість шифрів показано, що беззбитковість алфавіту, який використовується для представлення повідомлення, дозволяє забезпечити високу криптографічну стійкість, на підставі чого можна розділити шифри на досконалі і ідеальні;

7. В результаті розгляду методів криптоаналізу показана важливість використання методів стиснення для зменшення надмірності відкритого тексту і обсягу займаної інформації, в зв'язку з чим пропонується використання методів стиснення для вирішення проблеми захисту мовних сигналів;

8. Завданням дослідження є розробка спеціалізованого алгоритму захисту мовних повідомлень, використання параметризації для збільшення ефективної довжини ключа, та вивчення впливу параметризації на стійкість алгоритму, сформульовано вимоги до розроблених алгоритмів.

2. СТИСНЕННЯ МОВНИХ СИГНАЛІВ НА ОСНОВІ ДЕЛЬТА-ПЕРЕТВОРЕННЯ ДРУГОГО ПОРЯДКУ

2.1. Про оптимізовані дельта-перетворення другого порядку та стиснення інформації

При розробці методу стиснення мовної інформації ставиться проблема зменшення обсягу даних, необхідних для подальшого відновлення сигналу із заданою точністю [1]. Основна мета полягає в розробці алгоритмів стиснення, що відповідають вимогам застосунків, для яких вони розробляються. Крім ступеня стиснення, критеріями прийнятності того чи іншого методу кодування є складність алгоритмів кодування і декодування мови, ступінь відповідності між відновленими і вихідними сигналами, швидкість роботи алгоритмів кодування і декодування, можливість розширення і оптимізації методу і т.д. [6].

Вимоги сучасних додатків у багатьох випадках накладають обмеження на використання тих чи інших методів кодування мови з втратами. У той час як більшість існуючих методів кодування на основі ДІКМ і D-перетворень у багатьох випадках або не відповідають вимогам сучасних додатків по співвідношенню ступеня стиснення і точності відтворення, то найбільш серйозним обмеженням поширення алгоритмів стиснення, заснованих на кодуванні з перетворенням, як і раніше залишається висока обчислювальна трудомісткість і низька швидкість кодування/декодування. Для того щоб відтворити мову, часто необхідно погіршити якість відтворюваного сигналу і зменшити його розміри. Одним із шляхів вирішення цієї проблеми є розробка і впровадження більш досконалих апаратних засобів для декомпресії звукових послідовностей. Однак висока вартість, адаптивність лише до одного формату даних та пов'язані з ними обмеження та незручності мають значний вплив на можливості використання апаратних засобів у сфері кодування мовлення.

Альтернативою апаратному вирішенню проблеми є розробка і реалізація нових алгоритмів кодування мовних сигналів, які мають відносно невелику трудомісткість і відповідають вимогам більшості існуючих додатків за ступенем

і якістю стиснення мовної інформації. Одним з найпростіших методів обробки мови є D-перетворення, яке характеризується простотою, високою швидкістю і відносно низькою обчислювальною складністю. Для забезпечення достатньо ефективного кодування за співвідношенням ступеня стиснення і точності відтворення доцільно використовувати розроблені алгоритми оптимізованих D-перетворень другого порядку, які мають якісно кращі характеристики в порівнянні зі своїми попередниками [6]. Гнучкість і простота алгоритмів дозволяє адаптувати їх до завдань стиснення сигналів певної природи - мовних, біомедичних, відеосигналів [6].

У роботі використані модифіковані алгоритми D-перетворень, побудовані на основі двійкового алгоритму D-перетворення другого порядку, оптимізованого за точністю і швидкістю, вперше описаного в роботі [6].

2.2 Алгоритм двійкового дельта-перетворення зі згладжуванням

D-перетворення є способом побудови (апроксимації) для модульованої функції *y(t)* в дискретні моменти зміни незалежної змінної t_i відповідних наближень (демодульованих) значень $Y_i\left(t_i = t_0 + i * \nabla t\right), i = 0,1,2,...,t_0$ – початкове значення *t,* ∇t — крок перетворення, *i* — номер кроку, $y_i=y(t_i)$, $Y_i = Y(t_i))$. У той же час значення різниць в інтервалі перетворень приймають значення тільки з деякого обмеженого набору величин. Вони являють собою константи модуля і розрізняються за знаком кванти модуляції, що представляють собою сигнал ступінчастого вигляду, тобто змінюється в кожен тактовий момент часу на ∇Y_i.

Нижче наведено модифікований алгоритм D-перетворення другого порядку зі згладжуванням, який ефективно функціонує при наявності зовнішніх збурень $(\nabla Y 0)$ [6]. Згладжування виконується при реалізації алгоритму модуляції з використанням середнього значення приростів модульованої функції на інтервалі *[i, i+n]*. Алгоритми модуляції і демодуляції виглядають наступним чином [6]:

Кодер:

Модуляція:

$$\left.\begin{aligned} & z_i = Y_i - y_i; \\ & \nabla y_i^{згл} = \left(y_{i+1} - y_i\right)/n; \\ & \nabla z_i = \nabla Y_i - \nabla y_i^{згл}; \\ & F_i = z_i + 1{,}5\nabla z_i + \left(0.5\nabla z_i^2 / c - 0.125c\right) sign\left(\nabla z_i\right); \\ & \Delta_{i+1} = -sign\left(F_i\right); \end{aligned}\right\} \qquad (2.1)$$

Декодер:

$$\begin{aligned} & \text{Демодуляція} \\ & \nabla^2 Y_{i+1} = c * \Delta_{i+1}; \\ & \nabla Y_{i+1} = \nabla Y_i + \nabla^2 Y_{i+1}; \\ & Y_{i+1} = Y_i + \nabla Y_{i+1}; \quad c^* \geq c; c > 0. \end{aligned} \qquad (2.2)$$

де $y_i = y(t_i)$ – значення модульованої функції на i-му кроці в момент часу t_i, $i=1,2,...;\ t_i = t_0 + i\nabla t;\ \nabla t$ - постійний крок дискретизації інтервалу часу (незалежна змінна); Y_i – значення демодульованої функції на і-му кроці; z_i – помилка перетворення на і-му кроці; ∇z_i – приріст помилки; F_i – функція перемикання на і-му кроці; $\nabla^2 Y_{i+1}$ – друга різниця демодульованої функції (квант модуляції); $\nabla^2 Y_{i+1}$ – це перша різниця демодульованої функції; n - крок прогнозування; sign(x) $\in \{-1;+1\}$, причому sign(0)=+l або sign(0)= – l.

У стаціонарному процесі власна похибка дельта-перетворення не перевищує 1,25 с (с*=с).

Приклад перетворення на основі цього алгоритму для звукового сигналу (чоловічий голос) з частотою дискретизації 32 кГц для кроку прогнозування n = 5 наведено в таблиці 2.1 і проілюстровано на рис. 2.1 (с*=48)

Таблиця 2.1 – Таблиця значень D-перетворення

I	$\nabla^2 Y_i$	∇Y_i	Y_i	y_i	$\nabla y_i^{згл}$	z_i	∇z_i	F_i	Δ_i
0	48	48	-5897	-5945	-129,4	0	0	-4,5	1
1	-48	0	-5897	-6223	-64,8	326	112,8	667,4	1
2	-48	-48	-5945	-6416	-6,8	519	6,8	525,3	-1
3	-48	-96	-6041	-6534	46,4	589	-94,4	328,1	-1
4	48	-48	-6089	-6590	98	549	-194	-260,2	-1
5	48	0	-6089	-6592	149	503	-197	-327	1
6	48	48	-6041	-6547	199,4	458	-199,4	-388,8	1
7	48	96	-5945	-6450	244,8	409	-196,8	-419,6	1
8	48	144	-5801	-6302	283,4	357	-187,4	-407,4	1
9	48	192	-5609	-6100	312,4	299	-168,4	-343	1
10	48	240	-5369	-5847	332	238	-140	-239,7	1
11	48	288	-5081	-5550	344,6	181	-104,6	-123,4	1
12	48	336	-4745	-5226	355,2	145	-67,2	-14	1
13	-48	288	-4457	-4885	368,2	14П	-32,2	81,8	1
14	48	336	-4121	-4538	387,6	81	-99,6	-201,7	-1
15	48	384	-3737	-4187	414,2	66	-78,2	-131,7	1
16	48	432	-3305	-3827	446,4	90	-62,4	-53,2	1
17	-48	384	-2921	-3450	481	145	-49	42,7	1
18	48	432	-2489	-3044	513,6	123	-129,6	-300,2	-1
19	48	480	-2009	-2600	541,2	111	-109,2	-213,9	1
20	48	528	-1481	-2116	562	107	-82	-104,9	1
21	-48	480	-1001	-1595	576,8	114	-48,8	12,2	1
22	48	528	-473	-1045	587,2	44	-107,2	-271,9	-1
23	48	576	103	-476	595,8	3	-67,8	-158	1
24	48	624	727	106	603,2	-3	-27,2	-49,6	1
25	-48	576	1303	694	611	33	13	50,3	1
26	48	624	1927	1289	618	14	-42	-69	-1
27	-48	576	2503	1891	622,6	36	1,4	33,6	1
28	48	624	3127	2503	622,8	0	-46,8	-96,1	-1
29	-48	576	3703	3122	618	5	6	10	1
30	48	624	4327	3749	606,8	-46	-30,8	-100,9	-1
31	-48	576	4903	4379	588,6	-52	35,4i	14	1
32	48	624	5527	5004	564	-101	12	-85,5	-1
33	-48	576	6103	5617	531,6	-90	92,4	162,7	1
34	-48	528	6631	6212	491	-109	85	114,3	-1
35	-48	480	7111	6783	440	-152	88	83,1	-1
36	-48	432	7543	7322	379	-211	101	77,7	-1
37	-48	384	7927	7824	307,6	-281	124,4	116	-1
38	-48	336	8263	8275	227,4	-348	156,6	223	-1
39	-48	288	8551	8667	140,8	-404	195,2	413,5	-1
40	-48	240	8791	8983	51,2	-432	236,8	697,5	-1

Продовження табл. 2.1

I	$\nabla^2 Y_i$	∇Y_i	Y_i	y_i	$\nabla y_i^{згл}$	z_i	∇z_i	F_i	Δ_i
41	-48	192	8983	9217	-40,4	-426	280,4	1082,1	-1
42	-48	144	9127	9362	-133	-379	325	1571	-1
43	-48	96	9223	9412	-227	-285	371	2178,7	-1
44	-48	48	9271	9371	-324,2	-148	420,2	2930,1	-1
45	-48	0	9271	9239	-424	32	472	3829,7	-1
46	-48	-48	9223	9015	-524,8	256	524,8	4863,9	-1
47	-48	-96	9127	8697	-621,2	526	573,2	5944,6	-1
48	-48	-144	8983	8277	-707	850	611	6947	-1
49	-48	-192	8791	7750	-774,6	1233	630,6	7697,4	-1
50	-48	-240	8551	7119	-820,6	1672	628,6	8098,4	-1
51	-48	-288	8263	6391	-844,6	2160	604,6	8139,4	-1
52	-48	-336	7927	5591	-851	2672	563	7914,3	-1
53	-48	-384	7543	4742	-846,6	3185	510,6	7567,4	-1
54	-48	-432	7111	3877	-839	3666	455	7219,3	-1
55	-48	-480	6631	3016	-831,8	4095	399,8	6910,2	-1

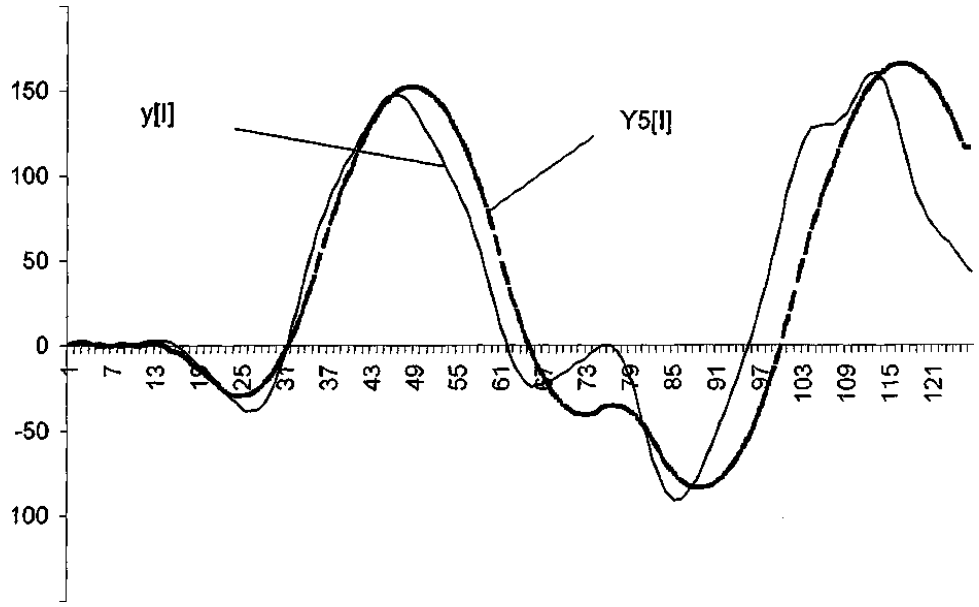


Рисунок 2.1 – Графіки модульованої функції *y[i]* та апроксимованої функції *Y5[i]* з використанням модифікованого алгоритму дельта-перетворення другого порядку зі згладжуванням (n=5)

Оцінки точності і продуктивності для алгоритму згладжування наведені в [11]. Наближеною функцією, зокрема, може бути послідовність цифрових вибірок мовного сигналу.

2.3 Використання D-перетворень для вирішення задачі стиснення мовних сигналів

Для підвищення точності кодування мовних сигналів більшість сучасних алгоритмів стиснення використовують в якості базової стратегії стиснення блокове (кадрове) перетворення [4], коли вхідні дані розбиваються на блоки, які потім перетворюються незалежно один від одного. Використання такої стратегії дозволяє поліпшити якість кодування за рахунок сталості параметрів сигналу в інтервалі перетворення і вирішити проблему завадостійкості при передачі стисненого сигнале по каналах зв'язку в цифрових системах зв'язку, так як помилка при передачі впливає тільки на блок, в якому це сталося.

У даній роботі при стисненні мовних сигналів на основі алгоритмів (2.1-2.2) використовується також стратегія блокового перетворення. Як видно (2.1-2.2), алгоритм є рядковим і, очевидно, може працювати з кадром будь-якої довжини. Для відновлення стисненого сигналу необхідними початковими умовами є: значення модульованої функції в початковий момент часу Y_0, значення ваги кванта модуляції c* за заданий інтервал і початкове значення першої різниці демодульованої функції ∇Y_0. Таким чином, кадр генерується кодером вихідного потоку повинен містити початкові умови і послідовність знаків кванта модуляції (D-послідовності). Кількість і частота початкових станів впливає на ступінь стиснення і якість кодування.

Аналіз алгоритмів (2.1-2.2) показує, що кожне з початкових умов по-різному впливає на демодульовану функцію. Тому для збільшення ступеня стиснення можна знехтувати впливом деяких початкових умов. Таким чином, початкове значення ∇Y_0 для кадру D-перетворення не робить істотного впливу на демодульованую функцію, тому припущення $\nabla Y_0=0$, що призведе лише до незначного зниження якості, але дозволить досягти більш високого ступеня стиснення. Відповідно, можливі такі стратегії формування стисненої послідовності: підвищення точності кодування і збільшення ступеня стиснення.

Для виконання вимог першої стратегії в даній роботі використовується наступний метод формування початкових умов: на кожен кадр $\nabla Y_0=0$, при поділі початкового сигналу на кадри, в блок початкових умов кожного кадру, значення $Y_{0,}c^*$ ($Y_0 = y_0$) для заданого кадру розрядна ширина початкових умов збігається з шириною біта вибірок вихідного сигналу.

Відповідно до другої стратегії початкові умови формуються наступним чином: при поділі на кадри початкове значення $Y_0 = y_0$ зберігається тільки для першого кадру, для наступних кадрів початкове значення демодульованої функції кадру k приймається як початкове значення демодульованої функції кадру $k-1$. $Y_0^k = Y_n^{k-1}$. Таким чином, блок початкових умов першого кадру містить значення $Y_{0,}c^*$, для наступних кадрів передається тільки c*. Для першого кадру використовується припущення $\nabla Y_0=0$, для наступних кадрів кадри об'єднуються і c* вибирається за методиками, описаними в [12].

В цілому обидві стратегії можна поєднувати з наступною моделлю: Оскільки значення c* має бути передане в обох випадках, будемо вважати, що c^* є службовою інформацією. Частоту передачі початкових умов кадру (Y_0) можна описати за допомогою періоду передачі початкового значення демодульованої функції (інтервалу перезапуску) U, яка може приймати значення від 1 до ∞ для сигналу з довжиною N кадрів. Значення $U=1$ відповідає передачі значення Y_0 у кожному кадрі, а значення $U=\infty$ відповідає наявності значення Y_0 тільки для першого кадру перетворення (збільшення ступеня стиснення). Використання значень U в діапазоні $1 < U < \infty$ тому можна розглядати як управління ступенем стиснення.

При стисненні мовної інформації за допомогою оптимізованих D-перетворень другого порядку необхідно враховувати, що велика довжина блоку негативно впливає на якість кодування, що обумовлено характеристиками джерела (мовного сигналу). У [9] зазначено, що параметри мовного сигналу залишаються постійними в інтервалі часу 10-20 мс, тому для частоти дискретизації 8000-44100 Гц цей інтервал становить приблизно 32-512 відліків. В результаті проведення експериментальних досліджень встановлено, що для

кодування мовних сигналів доцільно використовувати рамки довжиною від 64 до 192 відліків [14].

Одна з важливих проблем при стисненні на основі оптимізованих D-перетворень – це вибір ваги кванта модуляції c*. Як можна побачити з алгоритмів D-перетворення (2.1-2.2) значення c* впливає на точність апроксимації модульованої функції на даному кроці. Для того щоб досягти найкращої якості кодування, потрібен алгоритм розрахунку c*, що забезпечує мінімальну похибку кодування. Для обчислення c* можна використовувати кілька алгоритмів: на основі першої різниці, на основі другої різниці і т.д. вимагає додаткового часу на перегляд кадру вихідного сигналу, але дозволяє значно підвищити точність кодування. В результаті експериментальних досліджень було встановлено, що для розрахунку c* при стисненні мовних повідомлень доцільно використовувати алгоритм на основі усереднення значення другої різниці [14].

2.4 Статистичні властивості D-послідовностей і довжина повідомлення

З позицій ймовірності появи наступного символу («0» або «1») розглянемо розподіл вихідного потоку кодера (D-послідовність). Поява «1» або «0» в наступному біті стисненого потоку є випадковою величиною, так як визначається вхідним мовним сигналом, що представляє собою нестаціонарний випадковий процес [4]. Розглянемо, що для D-послідовності $p_0 \approx p_1 \approx 1/2$, то розподіл отриманої дельта-бітної послідовності (D-біт) близька до однорідної, і, відповідно, D-послідовність характеризується беззбитковістю. Оцінимо статистичні властивості D-послідовностей, припускаючи, що поява «0» і «1» рівноймовірна. Оскільки кванти модуляції приймають значення тільки з обмеженого набору {-1;+1} (або {0; 1} - при практичному використанні алгоритмів стиснення), кількість символів цього алфавіту дорівнює 2. Для D-послідовності. Можна оцінити математичне очікування значення *X (де X — поява* «0» або «1» у наступному кванті модуляції):

$$M\left(X\right)=0\cdot\frac{1}{2}+1\cdot\frac{1}{2}=\frac{1}{2}.$$

Очевидно, що закон розподілу величини X^2 буде співпадати з розподілом X:

X	0	1
P	0.5	0.5
X^2	0	1
P	0.5	0.5

Тоді математичне сподівання X^2

$$M\left(X^2\right)=0\cdot\frac{1}{2}+1\cdot\frac{1}{2}=\frac{1}{2}.$$

Дисперсія в цьому випадку складе [26]:

$$D(X)=M\left(X^2\right)-\left[M\left(X\right)\right]^2=\frac{1}{2}-\frac{1}{4}=\frac{1}{4}.$$

Можна показати, що для такого розподілу всі початкові моменти порядку k постійні:

$$v_k=M\left(X^k\right)=\frac{1}{2},$$

$$M\left(X^k\right)=0^k\cdot\frac{1}{2}+1^k\frac{1}{2}=\frac{1}{2}.$$

Для центральних моментів k-го порядку можна отримати наступні числові оцінки:

$$\mu_1 = M\left[X - M\left(X\right)\right] = 0,$$
$$\mu_1 = D\left(X\right) = \frac{1}{4},$$
$$\mu_3 = \nu_3 - 3\cdot\nu_1\cdot\nu_2 + 2\cdot\nu_1^3 = 0,$$
$$\mu_4 = \nu_4 - 4\cdot\nu_3\cdot\nu_1 + 6\cdot\nu_2\cdot\nu_1^2 - 3\cdot\nu_1^4 = \frac{1}{16}.$$

Ентропія алфавіту D-послідовності, що характеризує мінімальну кількість біт, необхідне для передачі одного символу алфавіту, буде:

$$H_\Lambda = -\sum_{i=1}^{n} p_i \cdot \log_2 p_i \approx -2\cdot\frac{1}{2}\cdot\log_2 2^{-1} \approx 1 - o\left(N\right),$$

де *o(N)* – мала величина, що характеризує відхилення від рівномірного розподілу для зв'язку довжини N.

Надмірність мови R_A для D-послідовності обчислюється як:

$$R_\Lambda = 1 - \frac{1 - o\left(N\right)}{\log_2 2} = 1 - 1 + o\left(N\right) = 0\left(N\right).$$

Таким чином, надмірність мови, що описує кванти модуляції, дорівнює малому значенню *o(N).* Надмірність мови криптограм є важливою характеристикою в криптоаналізі і обернено пропорційна відстані єдиності апріорного невизначеного шифру L:

$$L = \frac{\log_2 \left|K\right|}{R_\Lambda \cdot \log_2 n},$$

де K – ключ шифру, $|K|$ – потужність набору ключів.

Дистанція унікальності характеризує кількість символів криптограми, теоретично необхідних для однозначної розшифровки при необмежених ресурсах криптоаналітика. Таким чином, надмірність алфавіту і відстань сингулярності впливають не на обчислювальну стійкість шифру, а на кратність розв'язків криптограми. Якщо обсяг перехоплення менше відстані однозначності, то результатом розшифровки буде кілька рівноймовірних рішень, для яких важко зробити висновок, який з них вірний.

Для D-перетворень відстань єдиності дорівнюватиме:

$$L = \frac{\log_2 |K|}{R_\Lambda \cdot \log_2 n} = \frac{\log_2 |K|}{o(N)}.$$

Застосування стиснення сигналу на основі D-перетворень дозволяє збільшити дистанцію унікальності, так як для D-розрядного R_Λ це мала величина, значно менше, ніж в разі кодування інформації з багаторозрядними вибірками. Згідно з [9], для послідовностей такої мови використання простого алгоритму з малим ключем забезпечує високу стійкість. Тому застосування стиснення, заснованого на D-перетвореннях, крім зменшення об’єму мовного сигналу, забезпечує високу базову теоретичну стійкість (за рахунок наближення D-послідовностей заданого в алфавіті шифру до ідеальної) і тим самим визначає можливість використання простих алгоритмів захисту.

Використання D-перетворень для стиснення сигналу має значну перевагу перед іншими методами стиснення. Більшість існуючих алгоритмів стиснення інформації прагнуть досягти мінімальної довжини вихідної послідовності. Мінімальна довжина послідовності закодованих символів визначається з теореми К. Шеннона про вихідне кодування: мінімальним двійковим кодовим словом для кодування символу з ймовірністю p_i буде $\log_2 p_i$ двійкових розрядів. Для ненадлишкових джерел це значення буде постійним для всіх символів

алфавіту і не може бути зменшено, так як для такого джерела характерний рівномірний розподіл.

Таким чином, більшість методів стиснення прагнуть досягти мінімальної довжини кодового слова в середньому для всіх символів оригінального алфавіту, що досягається використанням перекосу розподілу як базової стратегії стиснення, тому для кодування символів, що часто зустрічаються використовуються коротші, ніж для рідкісних, через що інформація стискається. У цьому сенсі більшість існуючих методів стиснення не усувають частотні характеристики вхідного потоку, а лише стискають за рахунок використання змінного числа бітів для кодування різних символів алфавіту. Стиснення в деяких дельта-алгоритмах (таких як ДІКМ) досягається за рахунок зменшення кількості бітів, необхідних для зберігання різниці між відліками.

Відмінністю розглянутого алгоритму оптимізованого D-перетворення від інших алгоритмів стиснення інформації є стратегія стиснення з фіксованою довжиною кодового слова (1 біт) для кожного символу вхідного потоку. Таким чином, D-перетворення досить сильно усуває статистичну надмірність, властиву вхідним потокам даних. Саме ця властивість D-послідовностей дає можливість, крім дуже великої відстані унікальності, використовувати прості алгоритми для ефективного (і, відповідно, більш швидкого) захисту голосових повідомлень. Дистанція унікальності характеризує кількість символів, на основі яких теоретично можна передбачити допустиму (значущу) послідовність символів, що описують вихід джерела. Для D-перетворень ця величина значно вище, ніж для текстової інформації, що говорить про однорідність D-послідовності і складність побудови ефективної моделі критерію відкритого тексту.

Статистичне тестування використовується для виявлення статистичних залежностей і підтвердження однорідності розподілу послідовностей. Зокрема, найбільш поширеним методом є частотний тест k-грам, який заснований на підрахунку кількості *k-грам* алфавіту для довільного повідомлення.

2.5. Інвертування біта

Як видно з наведених вище алгоритмів, вихідна послідовність кодера D-перетворення є потоком квантів модуляції {+1 ,-1}. При зберіганні і передачі така послідовність (D-послідовність) перетворюється в двійкову форму для зменшення розміру, коли "-1" кодується "0". При реконструкції сигналу з бітового потоку наступне значення демодульованої функції залежить від значення попереднього зразка. попередні значення D-біта і ваги квантової модуляції, тому алгоритм демодуляції можна записати наступним чином (для випадку $\nabla Y_0 \neq 0$):

$$Y_i = Y_0 + \nabla Y_0 \cdot i + i \cdot c^* \cdot \Delta_1 + (i-1) \cdot c^* \cdot \Delta_2 + ... + c^* \cdot \Delta_i.$$

Така залежність є важливою властивістю D-послідовностей, оскільки, як видно, зміна навіть одного дельта-біта призводить до спотворення (руйнування) сигналу після цього D-біта. Величина похибки вставки при інверсії біта для демодульованої функції дорівнюватиме (по модулю):

$$| Err | = 2 \cdot c^*$$

Зміна і-го D-біта ($(1 \le i \le N)$) руйнує сигнал на інтервалі [i, N] і вносить d в кожний наступний відлік помилку величиною $| Err | = 2 \cdot j \cdot c^*$ (по модулю), де $(1 \le j \le N-i)$ (рис. 2.2).

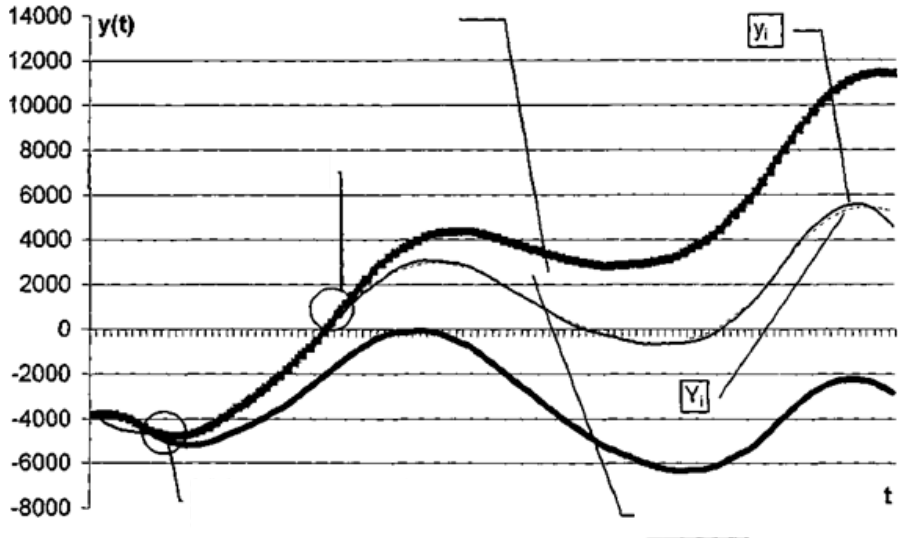


Рисунок 2.2 – Вплив інверсії бітів (Y_{inv}(-) - від "+1" до "-1", Y_{inv}(+) - від "-1" до "+1") D-послідовність на демодульовану функцію для сигналу мовлення з частотою дисретизації 32 кГц, крок прогнозування n = 5.

Ця залежність дозволяє ефективно використовувати інверсію бітів як один з алгоритмів захисту. У цьому випадку інверсія біта є комбінацією операцій перестановки і підстановки, так як кожен D-біт є однобітовим повідомленням і несе в собі інформацію про багаторозрядну вибірку вихідного голосового повідомлення. Роль «підстановки» виконує сама операція інверсії, але така «підстановка» визначена не для всіх символів алфавіту. Інверсія біта D-послідовності аналогічна операції інверсії спектру для захисту аналогових сигналів, однак для D-перетворення подальша бітова послідовність (аналог частотних діапазонів для методів скремблінгу) залежить від значення цього біта, що збільшує руйнування сигналу і ускладнює криптоаналіз.

Виходячи з цього, можна запропонувати використання операції інверсії групи бітів D-послідовності як складової частини алгоритму захисту мовної інформації на основі оптимізованих D-перетворень. При цьому, щоб зберегти рівномірність розподілу D-послідовності, доцільно використовувати інверсію парного числа бітів послідовності.

У загальному випадку можна оцінити інтервал руйнування сигналу при інверсії бітів для стратегій стиснення, розглянутих в розділі 2.3.

Для інтервалу перезапуску *U=1* початковий блок умов для кожного кадру містить власні значення Y_0,c*, тому інтервал руйнування сигналу обмежується змінною кадру. Інтервалом руйнування сигналу при інверсії біта є підпослідовність [i, n-i], де i - порядковий номер першого змінного біта, *n* - довжина кадру в бітах. Наступний кадр задасть свої початкові умови і міститиме справжнє значення сигналу. Щоб внести найбільшу похибку, необхідно змінити перші біти стисненого кадру.

Для $U = \infty$ блок початкових умов кадру (крім першого) не містить Y_0 і за початкове значення демодульованої функції попереднього кадру приймається як початкове значення демодульованої функції на інтервалі, тому інтервал руйнування сигналу при інверсії біта поширюється на весь сигнал після змінного біта. При цьому інтервал руйнування сигналу при інверсії біта дорівнює [i, M - i], де i - порядковий номер першого змінного біта, M - довжина сигналу в бітах. Введена похибка поширюється на всі наступні кадри і зростає зі збільшенням порядкових номерів кадру і D-біта.

Для $1<U<\infty$ інтервал руйнування сигналу - це інтервал перезапуску *U* (послідовність кадрів до передачі значення Y_0) [i, nT-i], де *i* - порядковий номер перевернутого біта, *n* - довжина кадру в бітах.

Очевидно, що на руйнування сигналу впливають і збережені початкові умови, тому зміна їх значень впливає на демодульований сигнал:

- Зміна значення Y_0 впливає на початкове значення демодульованої функції і еквівалентна зсуву сигналу по осі OY на величину введеної похибки.
- Зміна значення *c** впливає на величину приросту демодульованої функції і еквівалентне стиснення/розтягнення сигналу на величину введеної похибки.

Для *U=1* зміна Y_0 і c* впливає лише на кадр, в якому відбулися зміни, для $U=\infty$ - для всього наступного сигналу, для $1 < U < \infty$ - для інтервалу перезапуску.

Беручи до уваги перераховані вище міркування, можна стверджувати, що з точки зору інформаційної безпеки більш кращим варіантом формування початкових умов є використання частоти передачі початкових умов $U=\infty$, так як

навіть при незначних змінах вона забезпечує більшу деструкцію сигналу, яка зростає зі збільшенням порядкового номера кадру.

Беручи до уваги перераховані вище міркування і рівномірність розподілу D-бітів, інверсію D-послідовності в поєднанні з захистом початкового блоку умов доцільно використовувати в якості будівельних блоків для розробки алгоритму захисту мовних повідомлень на основі оптимізованих D-перетворень другого порядку.

2.6. Властивості D-послідовностей та використання криптографічних операцій для захисту D-послідовностей

Підводячи підсумок розгляду властивостей D-послідовностей, проведених в 2.4 і 2.5, можна відзначити, що D-послідовності з точки зору захисту мовної інформації мають наступні позитивні властивості:

- близький до рівномірного розподілу вихідної послідовності кодера, що значно збільшує дистанцію унікальності і не дозволяє побудувати ефективні критерії для відкритого тексту для D-послідовностей;
- мінімальна постійна довжина символів алфавіту (1 біт), що дозволяє стверджувати про виключення статистичних залежностей і, відповідно, в значній мірі знижує надмірність вхідного мовного сигналу;
- залежність значень демодульованої функції від попередніх D-послідовностей і початкових умов, що дозволяє використовувати бітову інверсію як складову частину розроблених алгоритмів захисту мовних повідомлень другого порядку на основі D.

У зв'язку з властивостями D-послідовностей для розроблених шифрів доцільно використовувати стійкі параметризовані операції для наближення розробленого алгоритму до ідеального і побудови гарантовано стабільного алгоритму.

Як було показано в попередніх розділах, розглянуті властивості D-послідовностей дозволяють використовувати більш прості алгоритми їх

криптографічного захисту. Однак «простота» алгоритму розглядається в криптографічному сенсі, тобто це не означає, що розроблений алгоритм повинен складатися з однієї операції на мові, так як в цьому випадку ми отримуємо «вироджений тип секретної системи», в якій єдиний ключ має ймовірність, рівну 1 і немає ніякої секретності системи. Тому необхідно розглянути доцільність використання криптографічних перетворень, що розглядаються в підрозділі 1.2, для захисту D-послідовностей і їх впливу на міцність.

Підстановка

Щоб застосувати операцію підстановки для захисту D-послідовностей, необхідно визначити одиницю виміру для операції підстановки – біт, бітовий вектор, байт тощо. Очевидно, що найпростішою заміною в цьому випадку буде заміна «0» на «1», а «1» на «0». Однак ця заміна є неефективною і має невелику тривалість, оскільки вона відповідає простій побітовій інверсії D-послідовності. Давайте обчислимо ентропію ключа і повідомлення для простої заміни, застосованої до мови з двобуквеним алфавітом, при цьому ймовірності для 0 і 1 дорівнюють p і q, а послідовні літери вибираються незалежно.

При цьому [12]:

$$H_E(X) = H_E(K) = -\sum P(E) \cdot P_E(K) \cdot \log P_E(K)$$

Імовірність того, що криптограма E довжиною N містить рівно s нулів у фіксованих місцях, дорівнює

$$p(E_s) = \frac{1}{2} \cdot \left(p^s \cdot q^{N-s} + q^q \cdot p^{N-s} \right),$$

і апостеріорні ймовірності однакових і обернених перестановок (тут тільки ці дві перестановки) рівні, відповідно

$$P_E(0)=\frac{p^s\cdot q^{N-s}}{\left(p^s\cdot q^{N-s}+q^s\cdot p^{N-s}\right)}, P_E(0)=\frac{q^s\cdot p^{N-s}}{\left(p^s\cdot q^{N-s}+q^s\cdot p^{N-s}\right)}.$$

Існує загальна кількість доданків C^s_N для кожного *s* і, отже:

$$H_E(K,N)=-\sum_s C_N^s\cdot p^s\cdot q^{N-s}\cdot\log_2\frac{p^s\cdot q^{N-s}}{p^s\cdot q^{N-s}+q^s\cdot p^{N-s}}.$$

Для p=1/3; q=2/3; p=1/8; q=7/8 величини H_E(K,N) показані на рис. 2.3.

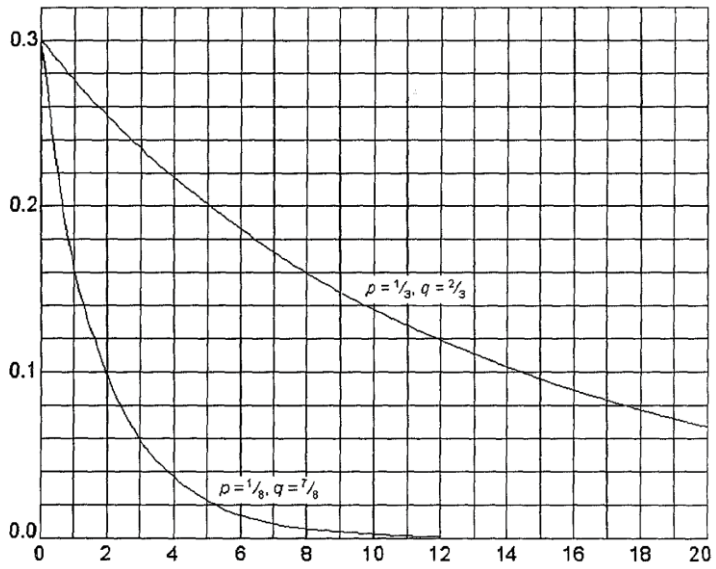


Рисунок 2.3 – Ентропія простої підстановки в двобуквеній мові

Як видно з рис.2.3 при нерівномірному розподілі ймовірностей *p* і *q* появи символів мови з двобуквеним алфавітом, для визначення використовуваної підстановки досить невеликого обсягу послідовності символів мови.

Враховуючи, що для D-послідовності p=q=1/2 ентропія дорівнюватиме

$$H_E(K,N)\approx-\sum_s C_N^s\cdot p^N\cdot\log_2\frac{p^N}{p^N+p^N}=\sum_s C_N^s\cdot p^N=2^N\cdot p^N=2^N\cdot\frac{1}{2^N}=1.$$

Таким чином, для D-послідовностей через рівномірність розподілу ентропія простої підстановки буде постійною величиною і буде описуватися на графіку прямою лінією. Однак число варіантів самої підстановки при використанні бітів в якості елементарних одиниць підстановки неприпустимо малт, тому при розробці алгоритму захисту D-послідовності необхідно використовувати підстановку для векторів більшої розрядної ширини.

Перестановка

Використання операції перестановки для захисту D-послідовності можливо 2 способами: перестановка за заданою таблицею перестановки необхідно також вирішити питання про елементарну одиницю перестановки. Очевидно, що недоцільно використовувати бітову перестановку для всього фрейму перетворення (який може мати довільну довжину), оскільки важко встановити та підтримувати таблицю перестановок, яка, з міркувань стійкості, не повинна бути однаковою для всіх кадрів. Операції перестановки для бітових векторів різної довжини залишаються майже тими ж проблемами, але розмір таблиці перестановок зменшується. У зв'язку з описаними недоліками табличної перестановки, більш відповідним варіантом використання цієї операції є застосування зсувів до стиснених даних, які просто визначаються аналітично і ефективно реалізуються на будь-якій платформі.

Гамма

Застосування гамма-хірургії для захисту D-послідовності має той самий ефект, що й при використанні операції інверсії групи бітів. При накладенні на D-послідовність довжини *N* гами шифру однакової довжини з допомогою операції XOR інвертує ті D-біти, які відповідають ненульовим розрядам шифрової гами, так як

$$\begin{cases} x \oplus 1 = \overline{x} \\ x \oplus 1 = x \end{cases}.$$

Тому ефективно реалізувати зміну групи бітів можна за допомогою операції XOR - замість зберігання номерів змінних бітів бітове поле з «1» доцільно зберігати в тих бітах, де необхідно інвертувати біт. Ви також можете використовувати XOR з виходом генератора PSP для захисту всього кадру, згенерованого кодером (початкові умови та D-послідовність). Оскільки D-послідовність має рівномірний розподіл (виконуються умови постановки ідеального шифру), то використання стабільного генератора PSP дозволить наблизити розроблений алгоритм до ідеального шифру.

2.7. Висновки

1. Розгляд алгоритмів стискання сигналів показав незадовільні швидкісні характеристики існуючих методів, тому для задачі стиснення мовних повідомлень пропонується використовувати високошвидкісні алгоритми стиснення та декомпресії на основі оптимізованих дельта-перетворень другого порядку;

2. Проаналізовано питання практичного використання D-перетворень, запропоновано використання стратегії блочного перетворення для стиснення мовних повідомлень, розглянуто варіанти формування початкових умов, запропоновано використовувати формування початкових умов кадру з інтервалом перезапуску *U,* зміна значення якого можна розглядати як керування ступенем стиснення;

3. Розглянуто характеристики D-послідовностей, що дозволяють стверджувати близькість розподілу D-бітів до рівномірного; отримано оцінки ентропії, надмірності та унікальності відстані для D-послідовностей, які дозволяють говорити про ефективність використання стиснення мовної інформації на основі D-перетворень для вирішення проблеми криптографічного захисту мовних повідомлень;

4. У результаті розгляду властивостей D-перетворень показано, що, на відміну від статистичних методів кодування, стиснення за допомогою D-

перетворень усуває статистичні залежності для вхідного мовного сигналу за рахунок мінімальної фіксованої довжини кодового слова (1 біт

5. Показано, що кожне значення демодульованої функції визначається початковими умовами і послідовністю значень D-біта, зміна розрядного значення призводить до спотворення наступних зразків демодульованого сигналу D-біта. Отримано оцінку величини введеної похибки спотворень для різних значень інтервалу перезапуску U. Запропоновано використання інверсії бітів як однієї зі складових алгоритму захисту. Розглянуто вплив зміни початкових умов на спотворення демодульованого сигналу для розглянутих значень U *;*

6. Сформульовано основні позитивні властивості D-послідовностей з точки зору захисту інформації: близькість розподілу D-послідовності до однорідної, що збільшує дистанцію унікальності для даного алфавіту, що призводить до відсутності альтернатив у відборі результатів криптоаналізу та не дозволяє побудувати ефективні критерії для відкритого тексту; Мінімальна фіксована довжина кодового слова (1 біт) для усунення статистичних залежностей вхідного потоку спотворення сигналу при зміні біта, що дозволяє використовувати цю операцію як складову частину розроблених алгоритмів захисту стиснених мовних повідомлень другого порядку на основі D-перетворень. Ці властивості дозволяють говорити про можливість створення криптографічного алгоритму для D-послідовностей, близького до ідеальних;

7. Для побудови надійного алгоритму пропонується використання криптографічно стійких перетворень для наближення розробленого шифру до ідеального;

8. На основі огляду елементарних криптографічних операцій та властивості D-послідовності проведено аналіз використання цих операцій для захисту D-послідовностей і робиться висновок про доцільність їх використання для захисту мовної інформації, стиснутої на основі оптимізованих D-перетворень другого порядку.

3. РОЗРОБКА АЛГОРИТМУ ЗАХИСТУ З ВИКОРИСТАННЯМ МОДИФІКАЦІЇ АЛГОРИТМУ RIJNDAEL

3.1. Обґрунтування вибору алгоритму

Відповідно до завдання захисту мовної інформації, властивостями D-послідовностей і вимогами до розробленого алгоритму в якості основного підходу в даній роботі використовується комбінація інформаційно-теоретичного і рандомізованого підходів.

Для задоволення вимог інформаційно-теоретичного підходу використовується стиснення цифрового голосового сигналу на основі оптимізованих D-перетворень, а для рандомізованого підходу використовується параметризація криптографічного алгоритму, запропонованого в 1.4. В даний час найбільш широко використовуваними архітектурами блочного шифру є мережі Фейстеля і узагальнені мережі підстановки і перестановки (мережі SP) (розділ 1.4).

Більшість сучасних блокових шифрів (DES і його модифікації, IDEA, LOKI, FEAL, Redoc, Khuni, Khafre, CAST, Blowfish) відносяться до узагальнених мереж Фейстеля, недоліками яких є низький рівень дифузії, велика кількість раундів для забезпечення прийнятної міцності і малі можливості в параметризації алгоритму, тому в даній роботі для параметризації була обрана інша архітектура побудови алгоритмів шифрування блоків.

На даний момент блочні шифри на основі поширених мереж SP є перспективним напрямком в криптографії [3]. Шифри такої архітектури мають такі позитивні характеристики:

- байт-орієнтована архітектура;
- наявність великих таблиць підстановки, що дозволяють ефективно реалізувати і можливість контролювати змішування.
- двофазне лінійне перетворення, що забезпечує дифузію змін відкритого тексту в двох напрямках - по рядках і стовпцях, що дозволяє

досягти сильної дисперсії в невеликій кількості раундів;

Прикладами шифрів цієї архітектури є криптографічні алгоритми SQUARE і Rijndael [4].

З огляду на недоліки алгоритмів, заснованих на мережах Фейстела, і перспективність розглянутої архітектури, для параметризації був обраний алгоритм Rijndael, затверджений в якості нового стандарту шифрування AES (Advanced Encryption Standard) [14]. Багаторічний аналіз Rijndael, проведений провідними світовими криптоаналітиками, не виявив жодних слабких місць в алгоритмі і підтвердив високі властивості змішування і розсіювання алгоритму, що послужило основою для вибору саме цього алгоритму для параметризації. Алгоритм відрізняється можливістю роботи з різною довжиною ключа і блоку, довжини яких можуть самостійно приймати значення 128, 192 і 256 біт. У зв'язку з вимогами, сформульованими в розділі 1 до алгоритмів захисту мовної інформації, стисненої на основі оптимізованих D-перетворень другого порядку, модифікований алгоритм на основі Rijndael повинен бути орієнтований на роботу з різною довжиною блоку і різною довжиною ключа.

3.2. Математична основа алгоритму Rijndael

Для перетворення інформації в алгоритмі використовується модульна арифметика в кінцевому полі (поле Галуа - GF(p), де p - просте число), в якому визначені операції додавання, віднімання, множення і ділення на ненульові елементи. Використання такого підходу обумовлено тим, що в ньому працює вся теорія чисел, в полі присутні числа тільки кінцевого розміру, похибки округлення при діленні відсутні, прямі обчислення прості, а зворотні – ні. З міркувань ефективності реалізації та відповідно до принципу байтової орієнтації обраної архітектури використовується поле GF(2), операції в якому мають деякі особливості. Оскільки деякі операції, визначені на рівні байтів, в розробленому алгоритмі відрізняються від загальноприйнятої двійкової арифметики, необхідно дати деякі пояснення.

Для представлення байта бітами $b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0$ використовується многочлен виду:

$$b_7 \cdot x^7 + b_6 \cdot x^6 + b_6 \cdot x^6 + b_5 \cdot x^5 + b_4 \cdot x^4 + b_3 \cdot x^3 + b_2 \cdot x^2 + b_1 \cdot x^1 + b_0$$

степені не вище 7 з коефіцієнтами {0,1}. Наприклад, значення 57_h (двійкове 01010111) представлено многочленом $x^6 + x^4 + x^2 + x + 1$. В якості операції додавання двох елементів в поле GF(2^8) використовується сума по модулю 2 коефіцієнтів з відповідними степенями: $57_h + 83_h = D4_h$.

У поліноміальній нотації:

$$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2$$

У двійковій арифметиці:

$$\begin{array}{r} 01010111 \\ \oplus \\ \underline{10000011} \\ 11010100 \end{array}$$

Очевидно, що додавання відповідає побітовій операції XOR на рівні байтів. При цьому кожен елемент має свою адитивну інверсію і тому операції додавання і віднімання еквівалентні.

У поліноміальному представленні множення в GF(2^8) є множенням двох многочленів за модулем P, де P – нескоротний многочлен степеня 8 (не має дільників, крім самого себе та 1). Додавання многочленів після множення здійснюється згідно з визначенням, описаним вище. Вибір многочлена P (простого числа) має досить значний вплив на результат. В описі алгоритму було обрано $P = x^8 + x^4 + x^3 + x + 1 = 01B_h$.

Наприклад, $57_h * 83_h = C1_h$, або:

$(x^6 + x^4 + x^2 + x + 1) \cdot (x^7 + x + 1) =$

$x^{13} + x^{11} + x^9 + x^8 + x^7 + x^7 + x^5 + x^3 + x^2 + x + x^6 + x^4 + x^2 + x + 1 =$

$= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$

$(x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3, +1) \bmod (x^8 + x^4 + x^3 + x + 1) = x^7 + x^6 + 1$

Результатом операції множення завжди є многочлен степеня нижче 8, що гарантує замкненість поля щодо множення. Таким чином, множення виконується не за правилами двійкової арифметики, що дещо ускладнює його виконання. Однак цілком можливо реалізувати множення за допомогою операції a=xt(b)=b*x.

У поліноміальному поданні операція визначається як:

$$xt(b) = (b_7 \cdot x^7 + b_6 \cdot x^6 + b_5 \cdot x^5 + b_4 \cdot x^4 + b_3 \cdot x^3 + b_2 \cdot x^2 + b_1 \cdot x^1 + b_0) \cdot x =$$
$$b_7 \cdot x^8 + b_6 \cdot x^7 + b_5 \cdot x^6 + b_4 \cdot x_5 + b_3 \cdot x_4 + b_2 \cdot x^3 + b_1 \cdot x^2 + b_0 \cdot x$$

Так як всі елементи поля є многочленами степеня нижче 8, то необхідно привести результат по модулю P. Якщо $b_7 = 0$, то перетворення не виконується, а якщо $b_7 = 1$, то виникає необхідність у віднімання P (тобто XOR). Цю операцію можна ефективно реалізувати шляхом зсуву вліво на 1 розряд і потім XOR за умови, що $b_7 = 1$. Множення на степінь x – це послідовне застосування до результату операції xt. Тоді можна ефективно реалізувати операцію множення на будь яке число: по двійковому представленні одного з множників виконується xt()з підсумовуванням проміжних результатів операцією XOR.

Операція ділення — це множення на обернений відносно множення поліном, такий, що $a(x) \bullet b(x) \bmod P = 1$, яке можна знайти за допомогою розширеного алгоритму Евкліда.

3.3. Опис алгоритму

За аналогією із загальноприйнятою в галузі криптографічного захисту термінологією введемо визначення основних елементів і понять пропонованого методу.

Стан - поточний результат перетворення, представлений у вигляді матриці байт з Num_Row рядками і Num_Col стовпцями. Кількість рядків і стовпців не може бути менше 4. Раундовий ключ шифрування також представлений у вигляді прямокутного масиву тієї ж розмірності, що і стан (рис. 3.1).

Вхідні дані позначаються як байти стану в порядку $a_{0,0}$, $a_{1,0}$, $a_{3,0}$, $a_{0,1}$, $a_{1,1}$, $a_{4,1}$,$a_{4,1}$, …, тобто стан заповнюється стовпцями. Після завершення перетворення вихідні дані отримуються з байтів стану в тому ж порядку.

Кількість циклів вказується як Num_Rounds і встановлюється користувачем. Для шифрів загальної архітектури мережі SP показано [4,14], що необхідні характеристики змішування і розсіювання досягаються за 2 раунди. Тому мінімальна кількість раундів не може бути менше 2, для Rijndael кількість раундів залежить від довжини ключа і коливається від 8 до 12, для розробленої модифікації рекомендується використовувати значення 4.

$a_{0,0}$	$a_{0,1}$	$A_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$
$a_{1,0}$	$a_{1,1}$	$A_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$
$a_{2,0}$	$a_{2,1}$	$A_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$
$a_{3,0}$	$a_{3,1}$	$A_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$	$k_{0,4}$	$k_{0,5}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$	$k_{1,4}$	$k_{1,5}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$	$k_{2,4}$	$k_{2,5}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$	$k_{3,4}$	$k_{3,5}$

Рисунок 3.1 – Приклад представлення станів (Num_Row=4, Num_Col=6) та раундового ключа

Перетворення раунду складається з чотирьох різних перетворень, які можна описати наступним чином (з використанням псевдокоду):

Раунд_Шифрування (стан, раундовий ключ)

Підстановка_байт (стан);

Зсув_рядків (стан)

Перемішування_стовпців (стан);

Гамування (стан, раундовий ключ)

Кожна з них реалізує обернене універсальне перетворення, яке називається шаром, а раунд, таким чином, складається з наступних шарів:

- Лінійний шар, що забезпечує збільшення дифузії вхідних даних зі збільшенням числа раундів (перетворення Зсув_рядків і Перемішування_стовпців);
- Нелінійний шар, що забезпечує нелінійність підстановки (Підстановка_байт);
- Гамма-шар, що складається з додавання раундового ключа до проміжного стану за допомогою операції XOR.

Підстановка_байт – це нелінійна підстановка, яка є незалежною для кожного байта стану. При впровадженні в програмне забезпечення вона виконується за допомогою підстановки таблиці для підвищення продуктивності. При проектуванні таблиць підстановки враховувалась стійкість до лінійного та диференціального криптоаналізу, з одного боку, та до інтерполяційних атак — з іншого [4,14].

Отримання оберненого елемента щодо множення в полі $GF(2^8)$ за модулем P, де P – нескоротний многочлен степеня 8 у полі $GF(2^8)$. При цьому нульовий і одиничний елементи перетворюються в самих себе.

Використання афінного перетворення, яке полягає в множенні елемента на матрицю з подальшим додаванням до вектора, над $GF(2^8)$ [14]. Використання цього перетворення необхідне для стійкості алгоритму до атак інтерполяції (SQUARE), які базуються на алгебраїчних властивостях скінченних полів і будують наближення таблиці підстановки. Результат цього перетворення можна представити у вигляді формули $S = AF(Inv(x))$, де x – байт стану, $Inv(x)$ –

мультиплікативна інверсія для елемента *x, AF(K)* – афінне перетворення. Для оберненого перетворення необхідно обчислити мультиплікативну інверсію афінного перетворення і отримати обернений елемент для результату.

У Rijndael було здійснено наступну трансформацію:

$$\begin{vmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ Y_7 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ X_7 \end{vmatrix} + \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix}$$

При цьому використовуване перетворення можна представити формулою:

$$AF(b_i) = b_i \oplus b_{(i+4)\bmod 8} \oplus b_{(i+5)\bmod 8} \oplus b_{(i+6)\bmod 8} \oplus b_{(i+7)\bmod 8} \oplus c_i,$$

де b_i – перший біт байта, c_i – відповідний біт константи-вектора. Для ефективної програмної реалізації Підстановки_Байт використовується підстановка таблиць на основі таблиць, попередньо розрахованих для фіксованих *p* і *AF(K)* за описаним алгоритмом 4 таблиць. Це вимагає деяких попередніх розрахунків і додаткового 1 КБ пам'яті (4 таблиці по 256 б кожна), але дозволяє значно збільшити швидкість.

Використання такого перетворення для побудови таблиці підстановки дозволяє легко розширити простір ключів за рахунок зміни основи відносно множення P і за допомогою другого афінного перетворення – для цього необхідно один раз повторити процедуру генерації таблиць підстановки з новими параметрами. Використання таблиць підстановки в складі ключа дозволяє збільшити довжину ключа з 256 до 610 біт. Крім того, запропонований

підхід дає можливість ефективно реалізувати сеансову схему зв’язку (за рахунок використання різних таблиць для кожної сесії).

Таким чином, невизначеність у виборі породжуючого поліному і афінного перетворення збільшує кількість варіантів переліку, що підвищує стійкість алгоритму в умовах апріорної невизначеності криптоаналітика щодо використовуваних параметрів перетворення.

Підстановка_Байт (Стан) застосовується послідовно до кожного байта стану.

Зсув_рядків (Стан) – це циклічний зсув вліво від кожного рядка стану на різну кількість байтів. Очевидно, що значення зсуву повинні бути більше 0 і менше Num_Col. У загальному випадку перший рядок залишається незмінним, а решта зсуваються на різну кількість байт для максимального перемішування. Значення зсуву задаються у вигляді таблиці і також можуть виступати в ролі частини ключа. Така стратегія вибору зсувів, крім досягнення дифузії вхідних даних підвищує стійкість до атак DCA і SQUARE (специфічно для шифрів такої архітектури), а також забезпечує простоту реалізації.

Значення зсуву для випадку квадратної матриці станів наведені в таблиці 3.1:

Таблиця 3.1. Величина зсувів для різної довжини блоку.

Кількість стовбців	Значення зсуву рядка					
	1	2	3	4	5	6
4	0	1	2	3	-	
5	0	1	2	3	4	
6	0	1	2	3	4	5

Таким чином, байт в позиції j в рядку і зсувається в позицію (j + Num_Col - C_i) mod Num_Col, де C_i - значення зміщення для і-го рядка.

Перемішування стовпців (стан) – це лінійне відображення вхідного вектора в Num_Row байтів на вихідний вектор тієї ж розмірності. Це перетворення необхідне для дифузії (перемішування) по стовпцях, тому що у запропонованому алгоритмі відкритий блок даних розглядається як двовимірний

масив і зміна одного біта вхідного потоку повинна впливати на кілька бітів зашифрованого тексту. Головними критеріями при проектуванні цієї трансформації були: оборотність, лінійність, симетрія, швидкість і простота. У цьому перетворенні стовпці станів розглядаються як многочлени в $Qp(2^{Num\text{-}Row})$ і множаться за модулем M (M – незвідний поліном степеня Num_Row) на многочлен c(x).

Коефіцієнти *c(x)* повинні бути ненульовими, в [14] для зручності виконання були обрані малі коефіцієнти. У даній роботі використовувалися поля $GF(2^4)$ і M=17, C(x) виглядає наступним чином:

$$c(x) = `03`x^2 + 01`x^2 + `01`x + `02`.$$

Це можна представити у вигляді множення матриць. Нехай *b(x)=c(x)a(x)*,

$$\begin{matrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{matrix} = \begin{matrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{matrix} \begin{matrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{matrix}$$

Додавання циклового ключа - це послідовний XOR байтів стану з байтами ключа циклу. З міркувань стійкості, розглянутих в розділі 1, довжина ключа циклу дорівнює довжині блоку і становить Num_Row*Num_Col байт. Схема генерації циклічного ключа заснована на використанні Byte_Substitution, досить складна і залежить від кількості раундів, докладний опис наведено в [14].

3.4. Розробка схеми управління ключами

Як було показано в розділі 1, переважна більшість алгоритмів використовує алгоритм створення ключів, який полягає у отримані послідовності циклічних ключів з майстер-ключа за більш-менш складним правилом. В даний

час це призводить до значного ускладнення схем розширення ключа і дуже великих витрат часу на розробку циклових ключів і додаткового споживання пам'яті для зберігання проміжних результатів, а також створює значні незручності при використанні ключів різної довжини, і, як наслідок, до помітного зниження продуктивності шифру.

Це пов'язано з тим, що на даний момент ключові схеми розширення досить складні в реалізації і повільні. Як видно з таблиць 3.3 і 3.4, час, витрачений на розширення ключа для одного блоку, значно перевищує час шифрування блоку (1 цикл відповідає 1 операції процесора [14]).

Таблиця 3.2 – Часові витрати на розширення ключа для одного блоку

Довжина блоку/ Довжина/ключа	Збільшення (шифрування), циклів	Збільшення (розшифровка), циклів	Шифрування, цикли
(128,128)	2100	2900	950
(192,128)	2600	3600	1125
(256,128)	2800	3800	1295

Як видно з таблиці 3.2, час, необхідний для розширення ключа, більш ніж в 2 рази перевищує час, необхідний для шифрування блоку, а також існує асиметрія часу, необхідного для подовження при шифруванні і дешифруванні (час отримання розширеного ключа шифрування для розшифрування набагато більше, ніж для шифрування).

З метою скорочення витрат часу на розширення ключа і збільшення швидкодії алгоритму пропонується використовувати композиційний варіант, що поєднує в собі принципи побудови блокового і потокового шифрів - використання генератора псевдовипадкових послідовностей (PSP) для генерації циклічних ключів, що дозволить збільшити швидкодію і стійкість системи в цілому. Перевагами використання генератора PSP є підвищена продуктивність схеми розширення ключа (і шифру в цілому), можливість реалізації потокового шифру, відсутність обмежень по довжині ключа шифрування, характерних для

існуючих блокових алгоритмів, а також можливість генерації ключа шифрування будь-якої довжини. Генератори PSP традиційно використовувалися для потокових шифрів, а існуючі блокові шифри не використовували генератори CSP для розширення ключа.

Конгруентні генератори PSP, регістри лінійного зсуву (LRS) та їх варіації, а також нелінійні генератори найбільш широко використовуються в криптографічних додатках.

Конгруентні генератори

Для генераторів цього класу можна зробити математично строгий висновок про властивості вихідних сигналів цих генераторів по періодичності і випадковості. Перевагою таких генераторів є простота програмної реалізації.

Лінійний конгруентний генератор виробляє послідовності псевдовипадкових чисел X(n), що описуються співвідношенням

$$X_n = (a \cdot X_{n-1} + c) mod\ M,$$

де a і c – константи.

Для роботи генератора необхідно встановити початкове значення (породжуюче число, повідомлення синхронізації) генератора X(0).

Лінійний конгруентний осцилятор ПСП видає псевдовипадкові числа з певним періодом повторення в залежності від обраних значень a і c. Датчик має максимальний період M, перш ніж згенерована послідовність почне повторюватися. Значення M зазвичай встановлюється рівним $2n$, де n - довжина машинного слова в бітах. Як показав Д. Кнут в [15], лінійний конгруентний генератор PSP має максимальну довжину M тоді і тільки тоді, коли c непарна і a mod 4 = 1.

Недоліком такого генератора є його передбачуваність і наявність добре відпрацьованих методів його відкриття. Розширенням лінійного конгруентного генератора є поліноміальні конгруентні генератори (квадратичні, кубічні та ін.), Робоче співвідношення яких виглядає наступним чином:

$$X_n = \left(a_1 \cdot X_{n-1}^k + a_2 \cdot X_{n-1}^{k-1} + ... + a_k \cdot X_{n-1} + c\right) \bmod M$$

Однак існують методи ефективного відкриття будь-яких поліноміальних генераторів з невідомими параметрами. Для використання таких генераторів в криптографічних додатках необхідно використовувати композицію з декількох генераторів для збільшення періоду PSP і складності атаки на генератор, основним недоліком такого підходу є складність і уповільненість реалізації.

Регістри лінійного зсуву

Регістри лінійного зсуву будуються з регістрів зсуву k-біт. На кожному такті вхідний біт зсуває k попередніх і до нього додається комбінація деяких бітів стану. Випереджаючий біт додається до гами (рисунок 3.2).

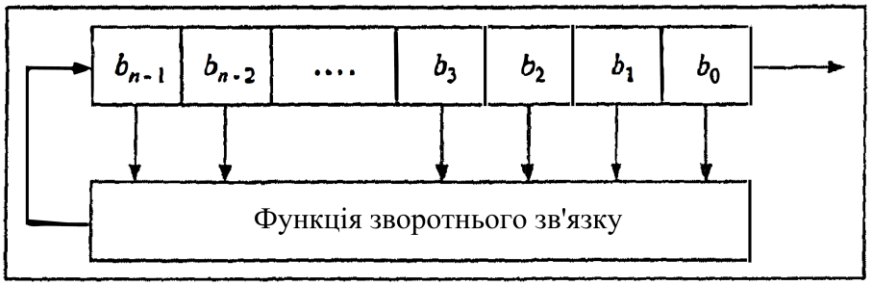


Рисунок 3.2 – Регістр лінійного зсуву

Період послідовності ЛРЗ, виходячи з її властивостей, дорівнює 2^k-1. Перевагою таких генераторів є простота виконання і легкість побудови каскадної схеми комбінуючого типу. ЛРЗ є найбільш вивченими генераторами ПСП, і цим пояснюється їх недолік – низька стійкість до злому. Регістри зсуву з лінійним зворотним зв'язком є розвитком ідеї ЛРЗ (рис. 3.3).

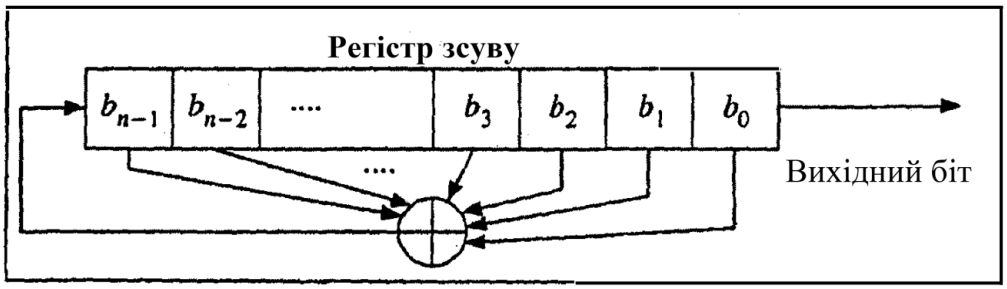


Рисунок 3.3 – Регістр зсуву з лінійним зворотним зв'язком

Загальним недоліком у використанні РЛЗ і РЗЛЗЗ є наявність добре розроблених відкритих методів апроксимації вихода такого генератора, наявність наближень і методик побудови узагальненого генератора ЛРС, авторство яких належить криптоаналітикам урядових криптографічних організацій, які настійно рекомендують використовувати генератори на базі ЛРЗ і РЗЛЗЗ. Як показано в [13, 14], РЛЗ і РЛЗЗ не відповідають всім тестам на випадковість PSP, що ставить під сумнів можливість використання таких генераторів при проектуванні шифрів.

Найбільш поширеними на даний момент є композитні генератори, що поєднують в собі генератори різних типів, так як багато в чому якість шифру, побудованого на основі генератора PSP, визначається як характеристиками датчика, так і алгоритмом отримання гамми. Перевагою композитних генераторів є відсутність добре відпрацьованих способів їх відкриття і складність застосування узагальнених підходів до відкриття. Основним недоліком є низький рівень дослідження їх властивостей, незначне зниження продуктивності і, найчастіше, складність апаратної реалізації.

Тому з міркувань стійкості, періодичності, випадковості та ефективності програмної реалізації в даній роботі був обраний генератор на основі множення з переносенням. [13]. Генератори множення на основі перенесення описуються наступним рівнянням:

$$x_n = a\,x_{n-1} + carry \bmod b.$$

Множник такого генератора – *a,* основа - *b,* а *carry* – значення перенесення. Для організації рекурсії необхідні початкові значення x_0 і *carry*, які і є ключом. Якщо початкове значення *carry* < *a,* то всі наступні переноси і сама послідовність будуть строго періодичними. Незважаючи на те, що загальна форма рівняння дуже схожа на рівняння лінійного конгруентного генератора, *carry* в даному випадку не є константою, тому властивості цих генераторів принципово різні.

Як показано в [14], умова максимального періоду для такого генератора - це простота чисел M і $\frac{M-1}{2}$, де $M = a \cdot b - 1$.

Для реалізації на процесорах Pentium в якості бази можуть бути обрані 2^{16}, 2^{32}, 2^{64}. З міркувань ефективності впровадження було обрано основу для цієї роботи $b = 2^{32}$.

Так як основа $b = 2^{32}$, то на виході генератора є 4-байтове ціле число, а для забезпечення мінімальної довжини раундового ключа в 128 біт потрібно всього 4 значення, а значить і 4 такта роботи генератора PSP. Так як у всіх операціях генератора використовується тільки цілочисельна арифметика, то запропонований генератор буде ефективно реалізований на будь-якій платформі з цілочисельною арифметикою.

Для реалізації схеми управління ключами пропонується використовувати такі режими:

1. Раундовий ключ - це сума за модулем 2 ключа шифрування і фрагмента PSP генератора МЗП, що дорівнює довжині оброблюваного блоку даних алгоритму;

2. Раундовий ключ являє собою фрагмент PSP генератора МЗП, що дорівнює довжині блоку даних алгоритму;

3. Для генерації раундового ключа використовується один з описаних режимів, причому використовуваний режим залежить від номеру раунду.

Перший режим роботи схеми управління ключами характеризується створенням нового генератора з невідомими параметрами, другий режим є найменш трудомістким, а третій режим є найбільш недетермінованим для

криптоаналітика в плані використовуваного ключа, але він характеризується ускладненням процедури управління ключами і необхідністю задання додаткового параметра для вказання, в якому раунді використовується генератор МЗП.

Описаний підхід до розширення ключа на базі генератора ПСП дає можливість використовувати будь-який генератор для генерації раундової ПСП.

Гнучкість запропонованого підходу до організації ключової схеми розширення полягає в простоті заміни одного генератора ПСП на інший, особливо в програмній реалізації. Цей підхід відповідає правилу Керкгофса, розглянутому в розділі 1, і має переваги перед оригінальною схемою розширення ключів Rijndael як з точки зору швидкості, так і гнучкості/масштабованості.

3.5 Оцінка впливу схеми параметризації на довговічність алгоритму

Як видно з опису алгоритму, в якості ключової інформації можуть виступати такі параметри алгоритму:

1. Формування многочлена для поля GF(2^8);

2. Множник і вектор афінного перетворення;

3. Поліном множення *c(x)* і основа *M* в перетворенні перемішування стовпців;

4. Таблиця значень зсуву в Зсуві_Рядків.

5. Параметри генератора МЗП (множник *a*, початкові значення x_0 і *carry*)**;**

6. Власне, ключ шифрування;

Щоб провести комплексну оцінку розробленого алгоритму, потрібно багато часу для того, щоб алгоритм був проаналізований авторитетними експертами. Однак, оскільки алгоритм є узагальненням [14], то для порівняння можуть бути використані його критерії стійкості.

Невизначеність у виборі породжуючого полінома дозволяє рандомізувати S-блоки алгоритму. Для GF(2^8) існує 42 простих числа, отже, існує 42 складові многочлени для перетворення Byte_Substitution. Ефект рандомізації для

побудови афінного перетворення можна оцінити наступним чином: припускаючи, що афінне перетворення задається наступною формулою: $AF(X) = A \cdot X + C$, де A - матриця, рядки якої представляють послідовні зсуви вектора породжуючого значення, а C - вектор-стовпець (для Rijndael обидва значення мають розмір 1 байт). Оскільки A і C вибираються незалежно, A≠0, можливо, 255*256 = 65280 варіантів побудови афінного перетворення. У поєднанні з вибором породжуючого полінома це дає 2 741 760 варіантів. Основа спепеня множення M може приймати п’ять значень, оцінка можливих значень $C\{x\}$ складна, оскільки, хоча коефіцієнти вибираються незалежно, результат множення за модулем може бути однаковим для різних коефіцієнтів $C\{x\}$ при однаковому вхідному сигналі. Для оцінки невизначеності таблиці зсуву можна використовувати той факт, що величина зсуву рядка не перевищує значення Num_Col, і всі рядки зміщуються на різні значення для забезпечення максимального розсіювання. Тоді кількість різних комбінацій значень зсувів рядків складатиме $Num\_Coll,$ а для рекомендованих значень для Rijndael 4,5,6 будуть 24, 120, 720 відповідно.

Параметри генератора МЗП можуть приймати будь-які ненульові значення, невизначеність яких визначає стійкість. Однією з умов їх вибору є максимальний період роботи генератора, але допустимо і вибір інших значень, так як це збільшує невизначеність параметрів. Для оцінки введеної невизначеності можуть бути використані рекомендовані значення множника a. Враховуючи, що значення не повинні перевищувати основу $b,$ для оцінки числа перебору початкових значень генератора з урахуванням всіх накладених обмежень можна прийняти значення b-1, що дорівнює $2^{32} – 1$.

Алгоритм Rijndael характеризується відсутністю більш ефективних методів злому, окрім алгоритму повного перебору [14] і відсутністю слабких ключів, тому модифікація схеми управління ключами і використання запропонованого генератора МЗП для генерації раундових підключів не знижує стійкість алгоритму. Так як мінімальна довжина блоку (як і довжина раундового ключа) становить 128 біт, то в кожному раунді довжина раундового ключа

збігається з довжиною перетворюваного блоку, стійкість алгоритму для атаки методом повного перебору становить близько 2^{128} операцій шифрування.

У цьому випадку можна оцінити вплив введеної параметризації на стійкість алгоритму як добуток можливого числа варіантів. Правомірність такої оцінки обумовлена тим, що для ДКА і ЛКА значення параметрів повинні бути відомі, тому навіть для цих методів криптоаналізу необхідний перебір значень параметрів. Тоді загальна кількість можливих значень параметрів (для 128-бітного блоку даних і ключа) складе:

$$N = N_P \cdot N_{AF} \cdot N_M \cdot N_{Shift} \cdot N_G = 42 \cdot 65280 \cdot 5 \cdot 24 \cdot (2^{32}-1) \approx 2^{60}.$$

Таким чином, введена параметризація збільшує довжину ключа мінімум в 1,5 рази і дає можливість отримати повний ключ криптографічного перетворення довжиною 2^{188} , що дозволяє говорити про доцільність запропонованого підходу.

Використання параметризації алгоритму з урахуванням беззбитковості стиснених голосових повідомлень дозволяє стверджувати, що для цього алгоритму не існує методів, більш ефективних, ніж атака методом повного перебору, яка в даний час обчислювально неможлива для 128-бітного ключа навіть з урахуванням розподіленої атаки [12].

3.7. Висновки

1. Для побудови надійного криптографічного алгоритму А1 для захисту стиснених D-перетворень мовних повідомлень другого порядку запропоновано використовувати комбінацію інформаційно-теоретичного та рандомізованого підходів;

2. В якості алгоритмів для задоволення вимог запропонованих підходів пропонується використовувати алгоритми на основі узагальнених мереж SP, які дозволяють забезпечити більшу параметризацію алгоритму. В якості прототипу для параметризації пропонується використовувати алгоритм Rijndael, так як він

дозволяє вводити більшу параметризацію і не містить слабких ключів;

3. З метою підвищення продуктивності алгоритму A1 за рахунок скорочення часу на генерацію раундових ключів, на відміну від традиційних схем розширення ключів, які характеризуються досить високою складністю, запропоновано використання генератора ПСП для генерації раундових ключів і розглянуто 3 можливих режими генерації раундових ключів;

4. Для підвищення невизначеності алгоритму, що використовується відповідно до вимог рандомізованого підходу, запропоновані параметри описаного алгоритму, невизначеність значень яких становить схему параметризації алгоритму криптографічного захисту;

5. Отримано оцінки впливу введеної параметризації на стійкість алгоритму A1 для захисту D-послідовностей, які показують можливість збільшення довжини повного ключа криптографічного перетворення A1 в 1,5 рази в порівнянні з Rijndael.

4. АЛГОРИТМ ЗАХИСТУ ГОЛОСОВИХ ПОВІДОМЛЕНЬ АЛГОРИТМ А2

4.1. Обґрунтування вибору використовуваних операцій

Дослідження в області теорії проектування блочних шифрів показали, що для побудови практичного стійкого блокового шифру необхідно забезпечити високий рівень змішування і розсіювання. Перемішування маскує зв'язок між відкритим текстом, зашифрованим текстом і ключем, і «... ускладнює статистику взаємозв'язків настільки, що навіть потужні криптоаналітичні інструменти стають непридатними». Розсіювання поширює вплив окремих бітів відкритого тексту на якомога більшу кількість зашифрованого тексту і маскує статистичні взаємозв'язки. Таким чином, більшість алгоритмів блочного шифрування містять комбінацію розсіювання та перемішування у вигляді раундової функції та реалізують ідею ітеративного блокового шифру, де проста функція раунду використовується кілька разів.

Стійкість багатьох шифрів безпосередньо пов'язана з використовуваними блоками розсіювання (S-блоками), які є відображенням *m*-бітових входів на *L*-бітові виходи - *m*n*-бітові S-блоки**.** Оскільки, як правило, обробка в S-блоках є єдиною нелінійною операцією в алгоритмі, то часто саме вони визначають стійкість алгоритму, яка лінійно залежить від їх розміру.

При проектуванні S-блоків необхідно зробити обґрунтований вибір *m* та *n*, так як збільшення *n* знижує ефективність диференціального криптоаналізу, але значно підвищує ефективність лінійного криптоаналізу [9]. Існуючі блочні алгоритми використовують S-блоки наступних розмірів: 6*4 (DES), 8*32 (Khufii, Khafre, Blowfish, CAST), 12*8 (LOKI), 16*16 (IDEA), 8*8 (AES). Для алгоритму, який розробляється D-послідовностией, доцільно використовувати S-блок з розміром входу не менше 8 для стійкості диференціальному і лінійному криптоаналізу. Врахування вимог щодо відповідності розміру відкритого тексту і зашифрованого тексту і зменшення обсягу пам'яті, що використовується для

зберігання таблиці, призводить до використання симетричної таблиці підстановки. Тому для розробленого алгоритму представляється доцільним використовувати S-блок розміром 8*8, в якому реалізована підстановка байтів. Розмір таблиці в даному випадку становить 256 байт.

Існує багато конкуруючих підходів до вирішення проблеми проектування S-блоків, але єдиних формальних критеріїв проектування ще не сформувалося. В результаті розгляду існуючих методів побудови найбільш перспективним слід визнати побудову S-блоку на основі афінного перетворення, відмінними рисами якого є простота виконання і складність апроксимації, а також гнучкість використання параметрів перетворення.

Відповідно до властивостей D-послідовностей (руйнування сигналу при інверсії біта), цю особливість D-послідовностей доцільно використовувати для розробленого алгоритму. Згідно з 2.6, ефективна реалізація інверсії бітів можлива за допомогою гамма-операції. Так як розроблений алгоритм орієнтований на захист кадрів D-послідовності, то для захисту всього кадру, утвореного кодером (початкові умови і D-послідовність), доцільно використовувати генератор PSP. З міркувань стійкості, періодичності, випадковості та ефективності програмної реалізації пропонується використовувати генератор множення з перенесом (МЗП)

4.2. Опис алгоритму

За аналогією із загальноприйнятою в галузі криптографічного захисту термінологією введемо визначення основних елементів і понять пропонованого алгоритму.

Блок – поточний результат перетворення, представлений у вигляді лінійного масиву кадрів сигналу. Довжина блоку становить M кадрів. Кадр – це структура даних, що містить блок початкових умов для одного з розглянутих варіантів формування початкових умов (C1 або C2) та послідовність D-бітів.

Довжина кадру становить *L* біт, що включає в себе довжину блоку початкових умов *S* бітів і довжину D-послідовності з *N* бітів:

$$\boldsymbol{L = S + N.}$$

Стан - поточний результат перетворення блоку. Стан заповнюється відповідно до порядку кадрів вихідного сигналу. Після завершення перетворення вихідні кадри витягуються зі стану в тому ж порядку.

Виходячи з розгляду властивостей D-послідовностей і застосовності до них криптографічних операцій, може бути запропонований наступний алгоритм захисту голосових повідомлень на основі оптимізованих D-перетворень.

Раундові перетворення складається з чотирьох різних перетворень, які можуть бути описані за допомогою псевдокоду наступним чином:

Раунд_Шифрування (стан, раундовий ключ):

Для i=l до M

Інверсія_біт_Кадру (i, Раундовий ключ);

Перестановка_кадрів (Стан, Раундовий ключ);

Для i=1 до M

Гамування (i, Раундовий ключ);

Підстановка_байт_кадра (i, Раундовий ключ (Модуль P));

Раундовий ключ = Модифікація (Раундовий ключ);

Раундовий ключ містить позиції змінних бітів для всіх кадрів блоку (вектор *{FK}),* лінійний масив для перестановки кадрів (*вектор PK),* параметри генератора PSP *GK*, вектор, що задає підстановку *SK*.

Кожна з цих операцій реалізує оборотне універсальне перетворення, яке називається шаром, і раунд, таким чином, складається з наступних шарів:

- Лінійний шар, що забезпечує збільшення дифузії вхідних даних (перетворення Перестановка_кадрів);

- Нелінійний шар, що забезпечує нелінійність підстановки (Підстановки_байт_кадра);

- Гамма-шар, що складається з інверсії бітів D-послідовності (Inversion_Bit_Frame) і власне накладання гами шифру, що генерується за допомогою генератора PSP.

Інверсія-біт_кадра – гамма D-послідовності і-го кадру з *i*-им елементом вектора *{FK}, 1≤i≤M.*

Перестановка_кадрів реалізує перестановку кадрів сигналу всередині блоку відповідно до ключа перестановки ***PK***.

Підстановка_байт – це нелінійна підстановка, яка є незалежною для кожного байта стану. Таблиці замін будуються на основі композиції з двох перетворень:

1. Отримання оберненого елемента множення в полі $GF(2)^8$) по модулю *P,* де *P* - незвідний поліном степеня 8 в полі $GF(2^8)$ (тобто просте число менше 512 і більше 256). При цьому нульовий і одиничний елементи переходять в самі себе.

2. Застосування афінного перетворення, яке полягає в множенні елемента на матрицю з подальшим підсумовуванням з вектором, понад $GF(2^8)$ [14]: $S = A \cdot x + V$, де *x* - мультиплікативна інверсія елемента *x, A* - матриця бітова 8x8, *V* - постійна. Використання цього перетворення необхідне для стійкості алгоритму до інтерполяційних атак, заснованих на алгебраїчних властивостях скінченних полів.

Гамма - це накладення гами генератора PSP на весь кадр, що підлягає перетворенню (початкові умови, службова інформація і D-послідовність), що необхідно для підвищення довговічності алгоритму. В якості генератора ПСП пропонується використовувати генератор МЗП.

Модифікація необхідна для генерації нового раундового ключа. Після обробки блоку ключові елементи кадрів FK_i циклічно зсуваються вліво на c_F байти, а ключ РК циклічно зсувається на c_P байт вправо, що дає новий ключ для наступного кроку алгоритму.

Як ми раніше розглядали, необхідними якостями для практичного шифру є перемішування (для забезпечення статистичної рівномірності відкритого

тексту) і розсіювання (для максимального поширення змін відкритого тексту в шифротексті), для яких використовуються криптографічні операції перестановки (P-блоки) і підстановки (S-блоки).

Операція підстановки виступає в ролі розсіювача, а перестановка кадрів виступає в ролі перемішування вхідних даних. У цьому випадку ключовою інформацією алгоритму є вектор ключових параметрів $Key = (\{FK\},\{PKl\{SK\}c_F,c_B)$. Така параметризація алгоритму необхідна для внесення невизначеності в запропонований алгоритм і підвищення стійкості.

У розробленому алгоритмі для шифрування використовується один раунд (що складається з табличних підстановок, зсувів і перестановок) низької трудомісткості, що дозволяє говорити про можливість значного збільшення продуктивності в порівнянні з відомими алгоритмами, а розглянуті властивості D-послідовності дозволяють говорити про високу стабільність запропонованого методу захисту мовної інформації.

4.3. Обґрунтування вибору мінімальної довжини ключа

Обґрунтований вибір довжини ключа (і пов'язаного з ним розміру блоку) має сильний вплив на міцність шифру. Якщо розмір блоку невеликий, то блоковий шифр не може бути стійким, так як при відносно невеликому обсязі відомого вихідного тексту найбільш поширені значення вхідних блоків будуть вичерпані. В даний час використовуються шифри з розміром входу 64-256 біт. В загальному випадку, чим більше розмір блоку шифрування, тим більш висока стійкість досяжна, так як більший розмір блоку вимагає розгляду більшої кількості взаємодіючих бітів і більш різноманітних можливих алгоритмів перетворення, але занадто великий розмір блоку вимагає багато ресурсів як для апаратної, так і для програмної реалізації.

Тому, щоб вибрати розумну довжину ключа, необхідно оцінити стійкість алгоритму проти атаки методом повного перебору, яка на даний момент є найбільш небезпечною для блокових шифрів. Задача знаходження ключів

симетричної криптосистеми шляхом перерахування всіх можливих ключів відноситься до класу задач, що допускають розпаралелювання. Використання розподілених обчислень для організації перебору таких ключів дає можливість ефективно вирішувати трудомісткі завдання в цій області. Експоненціальне зростання продуктивності обчислень з плином часу (в 10 разів за 5 років) має ще більш значний вплив на загальне зростання продуктивності. Таким чином, прогрес у цій сфері можливий завдяки:

- використання науково-технічного прогресу і застосування технологічних нововведень для підвищення продуктивності окремого пристрою;

- збільшення кількості таких пристроїв в системі.

Інших способів збільшення обчислювальної потужності не існує. Тому для отримання оцінок мінімальної безпечної довжини ключа необхідно проаналізувати граничні значення цих двох тенденцій.

Перша оцінка максимальної продуктивності обчислювального пристрою пов'язана з визначенням максимальної швидкості, заснованої на фізичних законах нашого світу (максимальної швидкості передачі інформації і максимальної щільності запису інформації). В результаті аналізу максимальної продуктивності обчислювального пристрою можна отримати оцінку максимальної частоти процесора, що дорівнює $3 \cdot 10^{18}$ опера./сек. За 100 років безперервної роботи гіпотетичний процесор виконає приблизно 10^{28} операцій. За умови, що він зондує один ключ за один цикл своєї роботи, а розшифровка повідомлення на знайденому ключі відбувається миттєво, він зможе перебрати 10^{28} ключів, що є ключем довжиною 93 біта. При цьому передбачається, що обчислювальні ресурси процесорів не витрачаються на координацію їх взаємної роботи в системі. Виходячи з цього, можна зробити висновок, що ключ довжиною понад 93 біта є стійким проти атаки методом повного перебору (табл. 4.1).

Однією з проблем систем шифрування є відсутність гарантованої нижньої межі стійкості системи. Верхньою межею прийнято вважати потужність

множини ключів, а нижня межа відсутня, так як не виключена ймовірність випадкового визначення необхідного ключа.

Таблиця 4.1 – Довжина ключа в бітах для забезпечення необхідного степеню стійкості

	Кількість відсотків.	Оп/сек	Оп/рік	Довжина ключа	Оп/100 років	Довжина ключа
Один Процесор	1	$3*10^{18}$	10^{26}	86-87	1028	90
Інтернет (50 мільйонів) оперативників в секунду	$5*10^{7}$	$1,5*10^{26}$	$5*10^{33}$	111-112	$4,7*10^{35}$	118-119
Населення Землі	$5*10^{9}$	$1,5*10^{28}$	$5*10^{35}$	118-119	$4,7*10^{37}$	125-126
Земля	$5,3*10^{26}$	$1,6*10^{45}$	$5,1*10^{52}$	175-176	$5*10^{54}$	181-182

Таким чином, основним висновком даного дослідження є визначення мінімальної безпечної довжини ключа для розробленого алгоритму захисту D-послідовностей, що дорівнює 128 бітам, а подальше збільшення стійкості алгоритму можливо за рахунок рандомізованого підходу - введення в алгоритм параметризації.

4.4. Оцінка стійкості розробленого алгоритму захисту

Розглянемо стійкість запропонованого алгоритму захисту голосових повідомлень з позицій криптоаналізу. Частотний криптоаналіз запропонованого алгоритму в чистому вигляді малоефективний, так як для накладення гами і інвертування бітів використовується генератор ПСП. Неефективність частотного криптоаналізу випливає з розглянутих статистичних властивостей D-послідовностей (рівномірність розподілу), що не дозволяє побудувати ефективну модель джерела для відкритого тексту.

З точки зору диференціального криптоаналізу, для успішної атаки необхідно формувати пари «відкритий текст-шифротекст» з фіксованою різницею (різниця в даному випадку визначається за допомогою операції XOR), після чого проводиться поетапний аналіз зміни в процесі шифрування отриманих відмінностей, на підставі чого різним ключам призначаються різні ймовірності. У зв'язку з тим, що для генерації гами алгоритм використовує генератор МЗП, який надає новий ключ для наступного блоку відкритого тексту, аналіз відмінностей відкритого тексту для визначення ключа не застосовується, на підставі чого можна зробити висновок, що диференціальний криптоаналіз є неефективним у взлому запропонованого алгоритму.

З точки зору лінійного криптоаналізу, ефективність атаки багато в чому залежить від структури використовуваних S-блоків, а також орієнтована на визначення єдиного ключа для великої кількості пар відкритий текст-шифротекст. Згідно [4], використовувані в цьому алгоритмі S-блоки оптимізовані проти лінійного криптоаналізу, невизначеність у виборі породжуючого полінома (навіть без урахування афінного перетворення) також знижує ефективність такої атаки. Використання генератора МЗП для генерації гами постійно змінює гаму шифру (шуканий ключ в термінології ЛКА), що робить лінійний криптоаналіз неефективним для криптоаналізу запропонованого алгоритму. Можливо, запропоноване міркування не відповідає математичній строгості при побудові формальної задачі криптоаналізу, але сам криптоаналіз алгоритму шифрування є темою для окремого дослідження.

Тепер оцінимо стійкість алгоритму з точки зору атаки методом повного перебору. Верхня межа трудосткості для злому будь-якого шифру - це розмір блоку даних, який потрібно перетворити в бітах. Загальна довжина даних, що підлягають перетворенню, становить M·L біт. Очевидно, що верхньою межею стійкості слід вважати повне перерахування всіх можливих значень для блоку такої довжини. Отже, верхню межу довговічності алгоритму можна оцінити як:

$$W_{\sup} = 2^{M \cdot L}.$$

Щоб отримати нижню оцінку, потрібно врахувати кількість можливих варіантів перебору для кожного з параметрів алгоритму. Стійкість запропонованого методу, заснованого на інверсії біт, може бути оцінена наступним чином:

Згідно з принципом Керкгоффса передбачається, що криптоаналітик знає алгоритм шифрування, тому нижньою межею стійкості інверсії k біт для D-послідовності довжини N з урахуванням властивостей, розглянутих в розділі 2, можна вважати число комбінацій k елементів з N можливих:

$$W_{inv} = C_N^k = \frac{N!}{k!(N-k)!}$$

W_{inv} досягає максимуму при $k = \frac{N}{2}$, тому для збільшення стійкості доцільно використовувати інверсію $k = \frac{N}{2}$ біт. З цим значенням k значення W_{inv} можна порівняти з верхньою межею трудомісткості на відкриття W_{sup}.

Для кожного кадру існують W_{inv} варіантів інвертування біт, за умови, що вибір інверсії незалежний для кожного кадру для блоку з m кадрів, кількість варіантів буде $\{W_{inv})^m$. Число перестановок для m кадрів дорівнюватиме $m!$, а число можливих значень зсувів c_F і c_B дорівнює $(N-1)$ і $(M-1)$ відповідно. У полі GF(2^8) є 42 незвідних поліноми степеня 8 ступеня. Отже, без урахування варіантів побудови афінного перетворення, і припускаючи використання загальної таблиці підстановки, загальне число варіантів для перебору (за умови, що алгоритм відомий) будуть

$$W = 42 \cdot (N-1) \cdot (M-1) \cdot M!(W_{inv})^m.$$

Величину W можна розглядати як оцінку нижньої межі довговічності запропонованого методу.

Оскільки $W_{inv} = C_n^k = \frac{n!}{k! \cdot (n-k)!}$ при $k = \frac{n}{2}$ співрозмірна з 2^n (при n=128 різниця складає один порядок), то можна вважати, що $(W_{inv})^m \approx 2^{m \cdot n}$.

Для порівняльної характеристики значень числа переборів W_{inv} і W доцільно використовувати логарифмічну шкалу з основою 2. При цьому під значеннями W_{inv} і W мається на увазі кількість бітів, які необхідно шукати в середньому для визначення відкритих даних. Використання даної шкали обумовлено необхідністю порівняння з існуючими алгоритмами криптографічного захисту, стійкість яких (при відсутності більш ефективних алгоритмів криптоаналізу) вимірюється потужністю множини ключів.

Залежність W_{inv} від числа змінних бітів при різних довжинах кадрів перетворення і довжині блоку m=4 проілюстрована на рис.4.1:

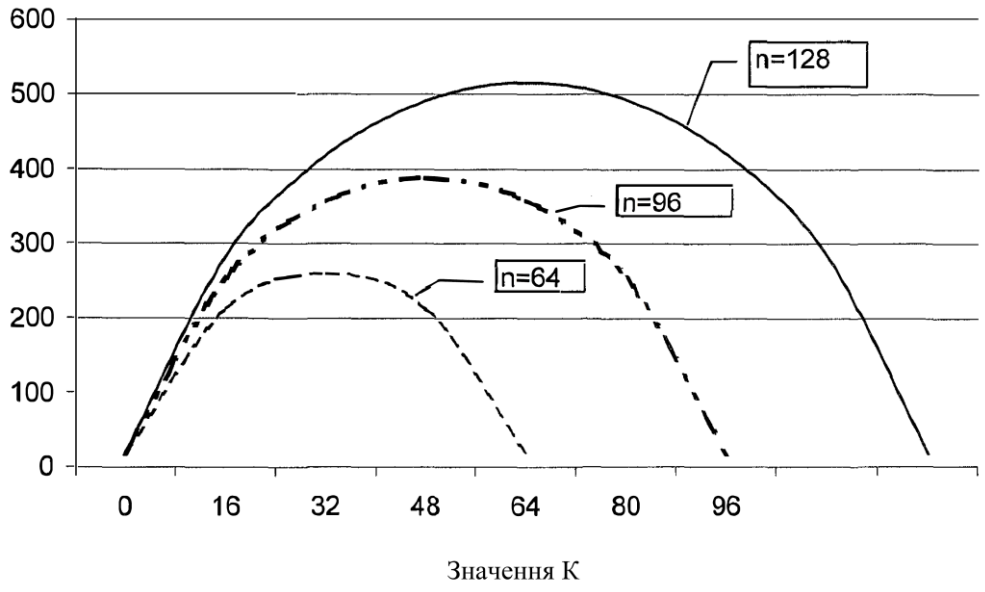


Рисунок 4.1 – Залежність W_{inv} для кадру перетворення довжиною n=64, 96, 128 і довжина блоку m=4.

Залежність опору W від довжини одиниці перетворення t в порівнянні з повним перерахуванням показана на рис.4.2

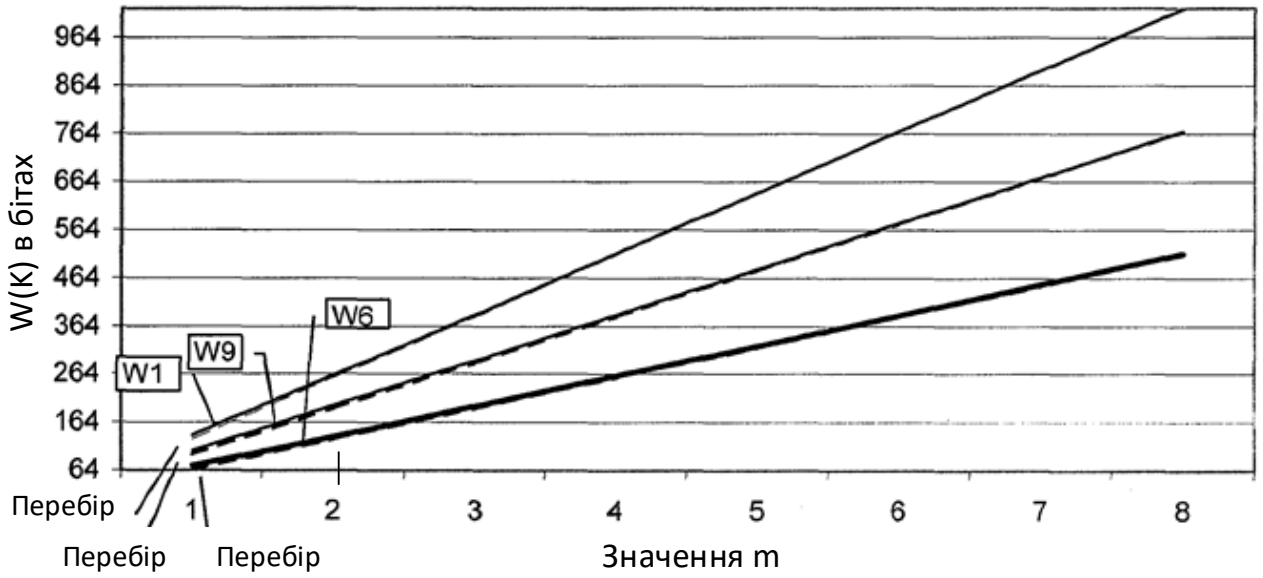


Рисунок 4.2 – Залежність *W* від *m* для системи перетворення довжиною n = 64, 96, 128 при *k=n/2*.

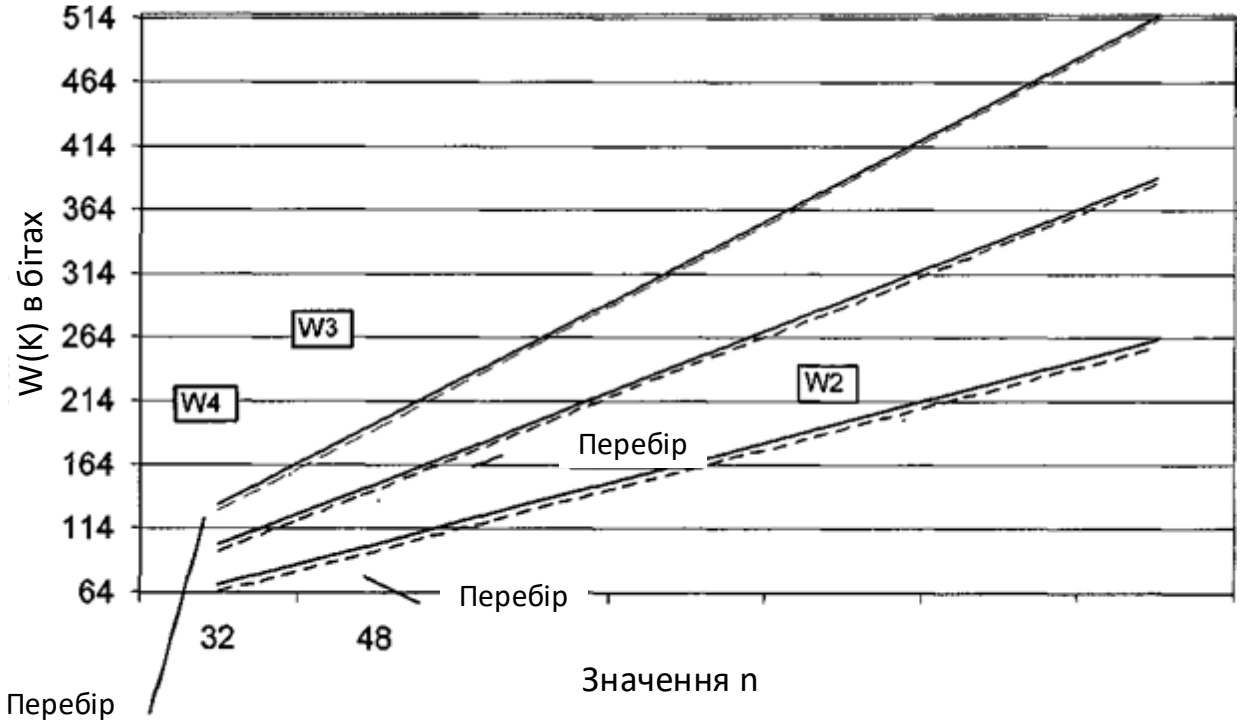


Рисунок 4.3 – Залежність *W* від *n* для одиниці перетворення довжини m=2,3,4 кадру при k=n/2.

Залежність стійкості W від довжини кадра n для блоку перетворення *m* в порівнянні з повним перебором показана на рис. 4.3.

Таким чином, як видно з рис.4.2, 4.3, отримана оцінка опору W при k=n/2 перевищує трудомісткість розкриття методом повного перебору.

Слід зазначити, що наведені вище оцінки не враховують складність відкриття використовуваного генератора МЗП, а ґрунтуються на властивостях D-послідовностей і вибраних криптографічних перетвореннях. Беручи до уваги, що ефективних методів відкриття для обраного генератора (принаймні у відкритих джерелах) не існує), можна зробити висновок, що запропонований алгоритм є перспективним (з міркувань надійності) для захисту голосових повідомлень з використання стиснення на основі оптимізованих D-перетворень другого порядку. Існуючі універсальні алгоритми блочного шифрування (DES, RC5, AES і т.д.) побудовані за ітераційним принципом і використовують велику кількість раундів (8, 12, 16, 32) для забезпечення достатньої надійності. У наведеному вище алгоритмі для шифрування використовується один раунд (що складається з підстановки таблиць, зсувів і перестановок) низької трудомісткості, що дозволяє говорити про можливість значного збільшення продуктивності в порівнянні з відомими алгоритмами при збереженні високої криптостійкості.

4.5. Оцінка продуктивності розроблених алгоритмів у порівнянні з існуючими блоковими шифрами

Важливим параметром будь-якого криптографічного алгоритму, крім стійкості, є його швидкодія, тому необхідно оцінювати швидкодію розроблених алгоритмів в порівнянні з існуючими ітераційними блоковими шифрами.

Для отримання об'єктивної оцінки продуктивності необхідно використовувати незалежні від реалізації характеристики алгоритмів шифрування, так як оцінка працездатності алгоритму в програмній реалізації сильно залежить від оптимізації програми і орієнтації на роботу з певною архітектурою процесора, операційною системою, довжиною блоку і т.д. елементарні операції, що складають цикл криптографічного перетворення (шифрування 1 відкритого текстового блоку). В результаті аналізу ітераційних

блокових шифрів в якості елементарних операцій для оцінки продуктивності пропонується використовувати такі операції:

- гамма (XOR) - *g;*
- сума - *s;*
- множення -*m;*
- зсув - *r;*
- таблична підстановка (S-блоки алгоритмів) - *t.*

Склад елементарних операцій, що використовуються для оцінки, визначається найбільшою поширеністю в існуючих блокових шифрах. Для бітової перестановки, яка зустрічається в деяких шифрах, пропонується використовувати операцію зсуву на число бітів перестановки в якості приблизної оцінки. Операції додавання і множення виконуються за певним модулем, який збігається з довжиною розрядної сітки, необхідної для представлення даних, якими оперує алгоритм: якщо алгоритм орієнтований на роботу з 16-розрядними даними то додавання і множення виконується за модулем 2^{16}, для 32-розрядних даних - за модулем 232 і т.д.

Таблиця 4.1 – Порівняльна характеристика продуктивності розроблених алгоритмів у порівнянні з існуючими блоковими шифрами в елементарних операціях.

Алгоритм	Продуктивність
DES	$DES = 640 \cdot r + 256 \cdot t + 32 \cdot g$
NewDES	$NewDES = 272 \cdot g + 136 \cdot f$
IDEA	$ІДЕЯ = 34 \cdot s + 34 \cdot m + 48 \cdot g + 7 \cdot r$
GOST	$GOST = 32 \cdot s + 256 \cdot t + 32 \cdot r + 32 \cdot g$
BlowFish	$BF_{KE} = 168 \cdot t + 384 \cdot s + 618 \cdot g$
$RC5_{16}$	$RC5_{16} = 265 \cdot s + 32 \cdot g + 230 \cdot r$
A1	$A1 = 128 \cdot t + l6 \cdot r + 4 \cdot g$
A2	$A2 = 68 \cdot t + 8 \cdot g + 5 \cdot r$

Однак слід зазначити, що така оцінка є приблизною і не враховує особливостей програмної реалізації порівнюваних алгоритмів (оптимізація етапів алгоритму, організація структур даних і т.д.). різна трудомісткість і, відповідно, різна кількість циклів процесора для виконання в залежності від типу процесора, розміру операндів і їх розташування. В якості максимальних характеристик для трудомісткості операцій доцільно використовувати такі характеристики: операції g і s займають ≈1 такт, операція множення m - від 13 до 42 тактів, операція зсуву r - від 1 до 5 тактів, таблиця підстановки t (розглядається як операція mov) - від 1 до 7 тактів (основою для таких характеристик є час, необхідний для виконання цих операцій процесором).

Відповідно до цього розглянемо вплив кожної з цих операцій на швидкодію розглянутих шифрів (рис. 4.4 - 4.6).

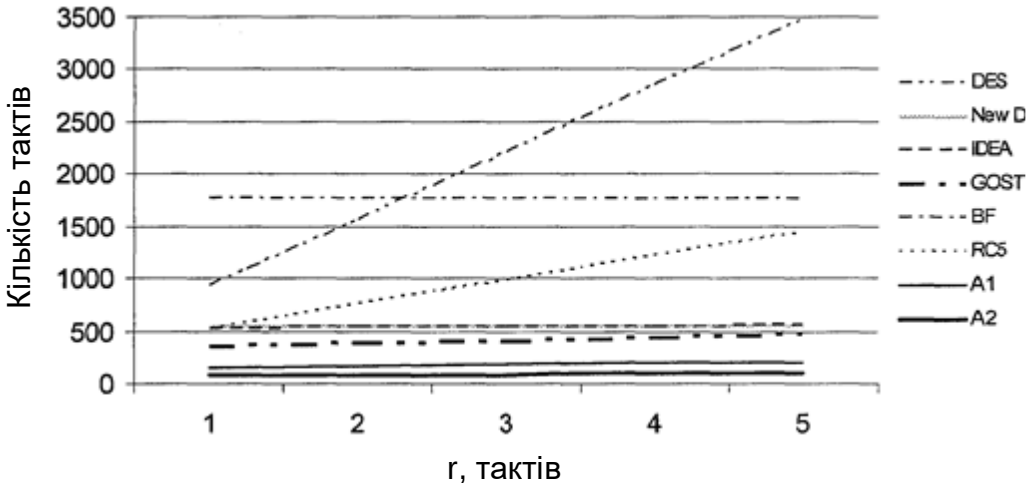


Рисунок 4.4 – Залежність продуктивності алгоритму від кількості тактів процесора, необхідних для виконання операції зсуву r (m=13, t=l)

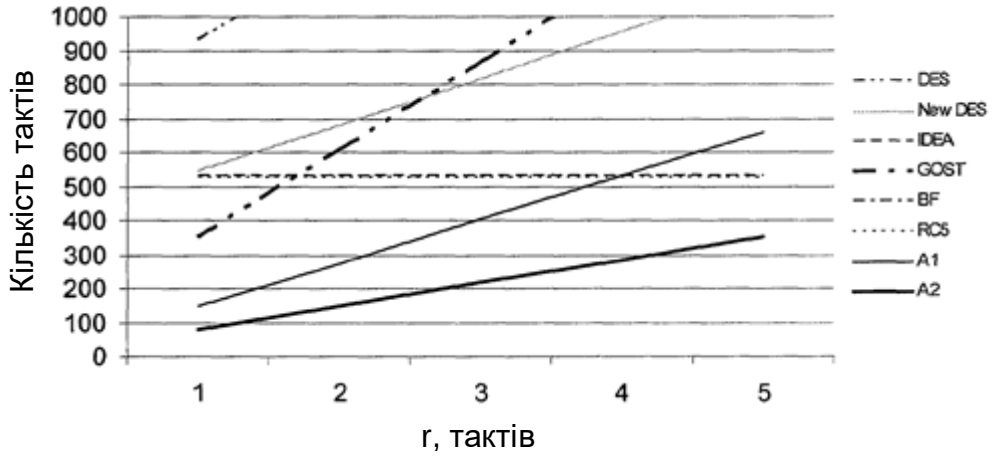


Рисунок 4.5 – Залежність швидкодії алгоритмів від кількості тактів процесора, необхідних для виконання операції пересилки t (m=13, r=3)

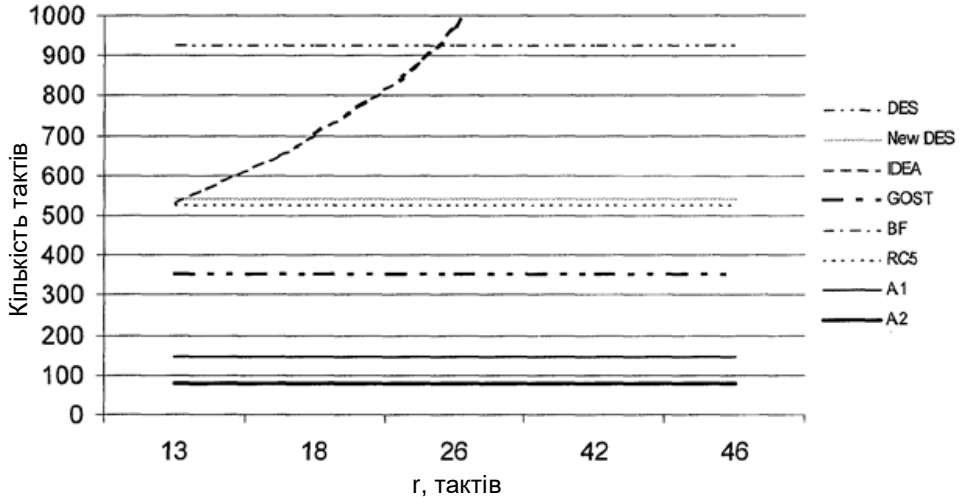


Рисунок 4.6 – Залежність продуктивності алгоритму від кількості циклів процесора, необхідних для виконання операції множення m (для t=l, r=3).

Як видно з таблиці 4.1 і рисунки 4.4-4.6, розроблені алгоритми А1 і А2 характеризуються досить високою швидкістю в порівнянні з існуючими аналогами за рахунок комбінації використовуваних операцій 128. Згідно з розглянутими оцінками, приріст продуктивності в порівнянні з розглянутими блоковими алгоритмами становить від 2,5 до 5 разів, при цьому розроблений алгоритм А2 приблизно в 1,5 рази швидше, ніж модифікований алгоритм А1.

Однак слід зазначити, що отримана теоретична оцінка є приблизною і не враховує особливостей програмної реалізації порівнюваних алгоритмів

(оптимізація кроків алгоритму, організація структур даних, особливості архітектури процесора і т.д.).

Для оцінки продуктивності програмної реалізації алгоритмів були проведені дослідження алгоритмів A1 та A2 у порівнянні з оптимізованими реалізаціями існуючих алгоритмів у бібліотеці Crypto++. Для перевірки швидкодії виконання алгоритмів використовувався комп'ютер з процесором Intel i5 під управлінням Windows 11. Результати дослідження наведені в таблиці 4.2 (в цьому випадку за результати тесту можна взяти r = 4, t = 4, m = 26).

Таблиця 4.2 – Порівняльна характеристика швидкодії розроблених алгоритмів у порівнянні з існуючими шифрами

Алгоритм	Розмір, байт	Час, сек.	Швидкість, Мбіт/сек
DES	134217728	9.94	12.87
NewDES	134217728	11.71	10.92
IDEA	134217728	11.28	11.34
RC5	268435456	11.08	23.11
BlowFish	134217728	7.09	18.05
GOST	134217728	12.83	9.98
Rijndael	268435456	8.44	30.32
A1	268435456	5.28	48.47
A2	268435456	4.18	61.24

Як видно з таблиці 4.2, алгоритми A1 і A2 характеризуються досить високою швидкодією. В експериментальних дослідженнях швидкодії алгоритмів A1 і A2 використовувалася неоптимізована реалізація алгоритмів, що пояснює меншу перевагу в порівнянні з очікуваною перевагою в порівнянні з існуючими шифрами. Тому, беручи до уваги можливу оптимізацію програмної реалізації алгоритмів A1 і A2, можна говорити про значне збільшення швидкодії розроблених алгоритмів в порівнянні з розглянутими блоковими шифрами.

Таким чином, розроблені методики характеризуються досить високими показниками швидкодії в порівнянні з існуючими алгоритмами, що дозволяє

говорити про доцільність і перспективи їх використання для захисту мовної інформації.

4.6. Висновки

1. Для побудови стійкого алгоритму на основі аналізу, проведеного в 2.6, були запропоновані стійкі операції підстановки, перестановки і гамма-перетворень;

2. На основі розглянутих властивостей D-послідовностей запропоновано алгоритм захисту мовної інформації A2, який містить бітову інверсію, перестановку, підстановку та зсуви, що забезпечують властивості розсіювання та змішування. Алгоритм містить один раунд, що дозволяє говорити про можливість збільшення продуктивності;

3. Для розробленого алгоритму A2, відповідно до принципу параметризації, запропоновано використання вектора параметрів криптографічного перетворення в якості ключа, що збільшує невизначеність ключа і, відповідно, підвищує міцність алгоритму;

4. Показано, що для запропонованого алгоритму захисту мовних повідомлень A2 зі стисненням на основі оптимізованих D-перетворень другого порядку найбільш ефективним алгоритмом відкриття є повний пошук;

Отримано верхню та нижню оцінки складності дешифрування розробленого алгоритму захисту голосових повідомлень, стиснених на основі оптимізованих D-перетворень.

5 ОХОРОНА ПРАЦІ

Охорона праці в Україні під час дослідження алгоритмів захисту аудіоінформації є основним аспектом забезпечення безпеки та ефективності робочого процесу. Діяльність, пов'язана з розробкою та тестуванням алгоритмів, забезпечує тривалу роботу з комп'ютерною технікою, обробку конфіденційних даних та, в окремих випадках, використання спеціалізованого обладнання. Відповідно до Закону України «Про охорону праці» (№ 2694-XII від 14.10.1992) та інших нормативно-правових актів, таких як ДСанПіН 3.3.2.007-98 і НПАОП 0.00-7.17-18, роботодавці зобов’язані створити безпечні умови праці, мінімізувати професійні ризики та забезпечити захист інформації. У цьому контексті ключовими аспектами є ергономічна організація робочих місць, електробезпека, захист від електромагнітного випромінювання та дотримання вимог щодо конфіденційності даних. Ці заходи спрямовані на збереження здоров'я працівників, підвищення продуктивності та забезпечення безпеки інформаційних ресурсів під час дослідження алгоритмів захисту аудіоінформації.

Дослідження на тему «Дослідження алгоритмів захисту аудіо інформації» відбувалася в приміщені, яке обладнане комп’ютеризованими робочими місцями. На дослідника мали вплив такі небезпечні та шкідливі виробничі фактори:

1. Фізичні: підвищена запиленість та загазованість повітря робочої зони; підвищений рівень шуму на робочому місці; підвищена чи понижена вологість повітря; підвищений рівень статичної електрики; підвищений рівень електромагнітного випромінювання; недостатня освітленість робочої зони.

2. Психофізіологічні: розумове перевантаження; перенапруга аналізаторів; статичне перевантаження.

Відповідно до визначених факторів формуємо рішення щодо безпечного виконання роботи.

5.1. Технічні рішення щодо безпечного виконання роботи

5.1.1. Обладнання робочого місця

Загальні ергономічні вимоги повинні враховувати характер і особливості трудової діяльності. [17].

Площа, виділена для одного робочого місця з ПК, повинна становити не менш 6 $м^2$, а об'єм – не менше 20 $м^3$ [3].

Природне освітлення має бути здійснене через світлові прорізи, спрямовані переважно на північ або північний схід, з метою забезпечення коефіцієнта природної освітленості (КПО) не менше, ніж 1,5%.

Виробничі приміщення повинні бути обладнані шафами для зберігання документів, стелажами, тумбами тощо, враховуючи вимоги до площі приміщень.

Приміщення з комп'ютерами потребують щоденного вологого прибирання.

Приміщення з комп'ютерами повинні бути оснащені аптечками першої медичної допомоги.

У приміщеннях з комп'ютерами мають бути розташовані приміщення для відпочинку під час роботи та кімната психологічного розвантаження. В кімнаті психологічного розвантаження слід передбачити встановлення пристроїв для приготування тонізуючих напоїв, а також місця для занять фізичною культурою.

Конструкція робочого місця користувача комп'ютера повинна забезпечити оптимальну робочу позу з такими ергономічними характеристиками: ноги - на підлозі або на підставці для ніг; стегна - у горизонтальній площині; передпліччя - вертикально; лікті - під кутом 70-90° до вертикальної площини; зап'ястя зігнуті під кутом не більше 20° щодо горизонтальної площини, нахил голови - 15-20° щодо вертикальної площини.

Висота робочої поверхні столу для відеотерміналу повинна знаходитися в межах 680-800 мм, а ширина - забезпечувати можливість виконання операцій у зоні досяжності моторного поля.

Робочий стіл для комп'ютера повинен мати простір для ніг з висотою не менше 600 мм, шириною не менше 500 мм, глибиною на рівні колін не менше 450 мм, на рівні витягнутої ноги - не менше 650 мм.

Робоче крісло для користувача комп'ютера повинне мати наступні основні елементи: сидіння, спинку та стаціонарні або знімні підлокітники.

Екран монітора й клавіатура повинні розташовуватися на оптимальній відстані від очей користувача, але не ближче 600 мм, з урахуванням розміру алфавітно-цифрових знаків і символів.

Клавіатуру варто розміщати на поверхні стола або на спеціальної, регульованої по висоті, робочій поверхні окремо від стола на відстані 100-300 мм від краю, найближчого до працівника. Кут нахилу клавіатури повинен бути в межах 5-15°.

При організації праці, пов’язаної з використанням ПК, для збереження здоров’я працюючих, запобігання професійним захворюванням І підтримки працездатності передбачаються внутрішньо змінні регламентовані перерви для відпочинку.

Внутрішньозмінні режими праці й відпочинку містять додаткові нетривалі перерви в періоди, що передують появі об’єктивних і суб’єктивних ознак стомлення й зниження працездатності.

Працюючі з ПК підлягають обов'язковим медичним оглядам: попереднім – при влаштуванні на роботу і періодичним – протягом трудової діяльності. Основними критеріями оцінки придатності до роботи з ПК мають бути показники стану органів зору: гострота зору, показники рефракції, акомодації, стану бінокулярного апарату ока тощо. При цьому необхідно враховувати також стан організму в цілому.

Лінія електромережі для живлення ПК, периферійних пристроїв ПК й устаткування для обслуговування, ремонту й налагодження ПК в приміщенні виконана як окрема групова трипровідна мережа, шляхом прокладання фазових, нульових робочих і нульового захисного провідників. Нульовий захисний провідник використається для заземлення електроприладів.

Металеві труби й гнучкі металеві рукави заземлені. Заземлення відповідає вимогам Правила безпечної експлуатації електроустановок споживачів [20].

Неприпустимим є:

– експлуатація кабелів і проводів з ушкодженими захисними властивостями за час експлуатації ізоляції; залишення під напругою кабелів і проводів з неізольованими провідниками;

– застосування саморобних подовжувачів, що не відповідають вимогам ПУЕ до переносних електропроводів;

– застосування для опалення приміщення нестандартного (саморобного) електронагрівального устаткування або ламп накалювання;

– користування ушкодженими розетками, вимикачами й іншими електровиробами, а також лампами, скло яких має сліди затемнення або здуття;

– підвішування світильників безпосередньо на струмоведучих проводах, обгортання електроламп і світильників папером, тканиною й іншими горючими матеріалами, експлуатація їх зі знятими ковпаками (розсіювачами);

– використання електроапаратури й приладів в умовах, що не відповідають вказівкам (рекомендаціям) підприємств-виготовлювачів.

5.2. Технічні рішення з гігієни праці та виробничої санітарії

5.2.1. Мікроклімат

Параметри мікроклімату нормуються в залежності від: періоду року; категорії робіт; технологічного процесу.

Для нормування параметрів мікроклімату календарний рік поділяється на два періоди:

– холодний період – період року, коли середньодобова температура зовні приміщення нижча за +10ºС;

– теплий – коли середньодобова температура зовні приміщення становить +10ºС і вище.

Робота розробника за енерговитратами відноситься до категорії 1а [21]. Допустимі параметри мікроклімату для категорії 1а наведені в табл.5.2.1 (відповідно до ДСН 3.3.6.042-99 [22]).

Таблиця 5.2.1 – Параметри мікроклімату

Період року	Допустимі		
	t,°C	W, %	V, м/с
Теплий	22-28	55	0,1-0,2
Холодний	21-25	75	0,1

Для забезпечення оптимального мікроклімату в приміщенні передбачено систему опалення та вентиляції повітря. Вимірювання показників мікроклімату планується здійснювати на початку, середині та в кінці холодного і теплого періодів року, не рідше трьох разів за кожну робочу зміну. У випадку коливань показників мікроклімату, пов'язаних з технологічними процесами або іншими факторами, виміри також плануються проводити при мінімальних та максимальних значеннях теплового навантаження на працюючих протягом робочого дня.

5.2.2. Склад повітря робочої зони

В приміщенні, де здійснюється розробка, можливими забруднювачами повітря може бути офісна техніка та пил, який потрапляє ззовні. ГДК шкідливих речовин, які знаходяться в досліджуваному приміщені, наведені в таблиці 5.2.2.

Таблиця 5.2.2 – ГДК шкідливих речовин у повітрі

Назва речовини	ГДК, мг/м^3		Клас небезпечності
	Максимально разова	Середньо добова	
Фенол	0,01	0,01	3
Формальдегід	0,035	0,003	2
Пил нетоксичний	0,5	0,15	4
Озон	0,16	0,03	4

В повітрі зовнішнього природного середовища, як і в повітряному середовищі приміщень завжди є наявною певна кількість заряджених частинок – іонів. Так в 1 см^3 чистого зовнішнього повітря міститься близько 1000 негативних іонів і понад 1200 позитивних. Параметри іонного складу повітря на робочому місці, що обладнане ПК, повинні відповідати допустимим нормам (табл.5.2.3).

Таблиця 5.2.3 – Рівні іонізації повітря приміщень при роботі на ПК

Рівні	Кількість іонів в 1 см^3	
	n+	n-
Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-5000
Максимально необхідні	50000	50000

Для забезпечення необхідного складу повітря в робочій зоні приміщення використовується система припливно-витяжної вентиляції. Постійно проводиться провітрювання за допомогою віконних отворів та регулярного вологого прибирання. Також передбачається встановлення системи кондиціювання.

5.2.3. Виробниче освітлення

Норми освітленості при штучному освітленні та КПО (для III пояса світлового клімату) при природному та сумісному освітленні для виконання роботи зазначені у таблиці 5.2.4 (відповідно до ДБН В.2.5-28-2018 [23]):

Таблиця 5.2.4 - Норми освітленості в приміщенні

Характеристика зорової роботи	Найменший розмір об'єкта розрізнювання	Розряд зорової роботи	Підрозряд зорової роботи	Контраст об'єкта розрізнення з фоном	Характеристика фона	Освітленість, лк		КПО, $e_н$, %			
						Штучне освітлення		Природнє освітлення		Сумісне освітлення	
						Комбіноване	Загальне	Верхнє або верхнє і бокове	Бокове	Верхнє або верхнє і бокове	Бокове
Дуже високої точності	Від 0,15 до 0,3	II	г	великий	світлий	1000	300	7	2,5	4,2	1,5

Для забезпечення комфортних умов роботи, рекомендується розташовувати робоче місце таким чином, щоб уникнути попадання прямого світла в очі. Щоб убезпечити відсутність світлових відблисків, використовують обладнання з матовою поверхнею. Захист очей від прямого сонячного світла або штучного освітлення можна забезпечити за допомогою захисних козирків та жалюзі на вікнах.

При створенні оптимальних умов для зорової роботи важливо враховувати не тільки кількість та якість освітлення, але й кольорове оформлення приміщення. Наприклад, якщо інтер'єр має світлі відтінки, то завдяки збільшенню кількості відбитого світла рівень освітленості підвищується на 20-40% (за тієї ж потужності джерел світла), а різкість тіней зменшується, що сприяє рівномірнішому освітленню.

5.2.4. Виробничий шум

Індивідуальні особливості людини, пов’язані з різними психологічними реакціями на вплив шуму, суттєво впливають на його сприйняття

Допустимі рівні шуму та вібрації на місцях праці осіб, що працюють з ПК, встановлені санітарними нормами ДсанПіН 3.3.2-007-98 [24], витяг з яких подано в таблиці 5.2.5.

Таблиця 5.2.5 - Допустимі еквівалентні рівні шуму

Вид професійної діяльності, місце праці	Еквівалентні рівні шуму, дБА.
Програмісти	50
Оператори в залах опрацювання інформації на ПК та оператори комп’ютерного набору	65
В приміщеннях для розташування шумних агрегатів	75

Головними методами зменшення впливу шуму є усунення або зменшення джерела шуму в процесі роботи, застосування матеріалів, що поглинають звук, і розумне організаційне планування приміщень виробництва.

5.2.5. Виробничі випромінювання

Оскільки дослідження проводилося за допомогою ПК, то на робочому місці працівника можливий підвищений рівень електромагнітного випромінювання.

Основою функціонування організму є дуже слабкі біоелектричні струми, що синхронізують природні біологічні режими. Штучні ЕМП якщо співпадають з частотами біологічних ритмів мозку або біоелектричною активністю серця чи інших органів людини можуть призвести до десинхронізації функціональних процесів в організмі.

Механізм біологічної дії на організм людини полягає як у тепловому, так і нетепловому специфічному ефекті, теплова дія ЕМП проявляються у

підвищенні температури тіла, а також локальному, вибірковому нагріванні тканин, органів, клітин унаслідок переходу електромагнітної енергії у теплову.

Допустимі значення параметрів неіонізуючих електромагнітних випромінювань від монітору комп'ютера представлені в табл. 5.2.6.

Таблиця 5.2.6 – Допустимі значення параметрів неіонізуючих електромагнітних випромінювань

Види поля	Допустимі параметри поля		Допустима поверхнева щільність потоку енергії (інтенсивність потоку енергії), Вт/м^2
	за електричною складовою (Е), В/м	за магнітною складовою (Н), А/М	
Напруженість електромагнітного поля, 6 кГц...3 МГц	50	5	
3 МГц...30МГц	2	-	
30 МГц...5 ГГц	-	-	10
Електромагнітне поле оптичного діапазону в ультрафіолетовій частині спектру: УФ-С (220...280 мм)			0,001
УФ-В (280...320 мм)			0,01
УФ-А (320. ..400 мм)			10,0
в інфрачервоній частині спектру: 0,76... 10,0 мкм			35,0.. .70,0
Напруженість електричного поля ВДТ			20 вВ/м

Для зменшення впливу електромагнітних полів від персональних комп'ютерів на дослідника, важливо дотримуватися встановлених нормативів робочого часу та регулярних перерв для відпочинку.

Для захисту від електромагнітних полів від ПК можна також застосовувати екранування, використовувати периферійні пристрої з низьким електромагнітним випромінюванням, а також встановлювати монітори з

низьким рівнем випромінювання, які відповідають відповідним стандартам безпеки.

5.3. Пожежна безпека

В приміщенні, де здійснювалось дослідження використовуються тільки негорючі речовини та матеріали у холодному стані, за ступенем вибухопожежної та пожежної небезпеки приміщення відноситься до категорії «Д».

За вогнестійкістю будинок відноситься до другої категорії [25]. Робоча зона приміщення віднесена до класу вибухонебезпечності В-IIа та пожежонебезпечності П-IIа, оскільки вибухонебезпечна концентрація пилу і волокон може утворюватися лише внаслідок аварії або несправності.

5.3.1. Технічні рішення системи запобігання пожежі

Система запобігання пожежі включає такі основні напрями:

– запобігання утворенню горючого середовища. Досягається: застосуванням герметичного виробничого устаткування; максимально можливою заміною в технологічних процесах горючих речовин та матеріалів негорючими; обмеженням кількості пожежо- та вибухонебезпечних речовин при використанні та зберіганні, а також правильним їх розміщенням; ізоляцією горючого та вибухонебезпечного середовища; організацією контролю за складом повітря в приміщенні та контролю за станом середовища в апаратах; застосуванням робочої та аварійної вентиляції; відведенням горючого середовища в спеціальні пристрої та безпечні місця; застосуванням в установках з горючими речовинами пристроїв захисту від пошкоджень та аварій; використанням інгібувальних (хімічно активні компоненти, що сприяють припиненню пожежі) та флегматизаційних (інертні компоненти, що роблять середовище негорючим) добавок та ін.

– запобігання виникненню в горючому середовищі (чи внесенню в нього) джерела запалювання. Досягається: використанням устаткування та пристроїв, при роботі яких не виникає джерел запалювання; використанням

електроустаткування, що відповідає за виконанням класу пожежо- та вибухонебезпеки приміщень та зон, груп і категорії вибухонебезпечної суміші; виконанням вимог щодо сумісного зберігання речовин та матеріалів; використанням устаткування, що задовольняє вимоги електростатичної іскробезпеки; улаштуванням блискавкозахисту; організацією автоматичного контролю параметрів, що визначають джерела запалювання; використанням швидкодіючих засобів захисного вимкнення; заземленням устаткування, видовжених металоконструкцій; використанням під час роботи з ЛЗР інструментів, що не допускають іскроутворення; ліквідацією умов для самоспалахування речовин і матеріалів; усуненням контакту з повітрям пірофорних речовин; підтриманням температури нагрівання поверхні устаткування пристроїв, речовин та матеріалів, які можуть контактувати з горючим середовищем нижче гранично допустимої (80 % температури самозаймання).

Можливі причини виникнення пожежі у приміщенні, де проводилося дослідження системи фазового автопідстроювання частоти такі:

– несправна електропроводка (іскріння, перегрів провідників, пересихання електроізоляційних матеріалів);

– залишення без нагляду увімкнутих комп’ютерів, обчислювальної техніки та інших електроприладів.

Для запобігання виникнення пожежі доцільні такі заходи:

– призначення осіб, що відповідальні за пожежну безпеку приміщення;

– систематичне проведення повторних протипожежних інструктажів та занять за програмою пожежно-технічного мінімуму з особами, що відповідальні за пожежну безпеку;

– утримання в справному стані засобів протипожежного захисту.

5.3.2. Технічні рішення системи протипожежного захисту

Протипожежний захист промислових об’єктів забезпечується:

– правильним вибором необхідного ступеня вогнестійкості будівельних конструкцій; правильним об’ємно-планувальним рішенням будівель і споруд; розташуванням приміщень та виробництв з урахуванням вимог пожежної безпеки;

– улаштуванням протипожежних перепон у будівлях, системах вентиляції, опалювальних та кабельних комунікаціях;

– спорудженням протидимного захисту;

– забезпеченням евакуації людей;

– використанням засобів пожежної сигналізації, сповіщування та пожежогасіння;

– організацією пожежної охорони об’єкта;

– засобами, що забезпечують успішне розгортання тактичних дій гасіння пожежі.

Досліджуване приміщення має два переносних (порошкових, водопінних або водяних) вогнегасників з масою заряду вогнегасної речовини 5 кг і більше. Крім того, треба передбачати по одному вуглекислотному вогнегаснику з величиною заряду вогнегасної речовини 3 кг і більше:

– на 20 м^2 площі підлоги в таких приміщеннях: офісні приміщення з ПЕОМ, комори, електрощитові, вентиляційні камери та інші технічні приміщення;

– на 50 м^2 площі підлоги приміщень архівів, машинних залів, бібліотек, музеїв [26].

ВИСНОВОК

У бакалаврській роботі проведено огляд методів захисту аналогових та цифрових повідомлень, що дозволяє стверджувати про відсутність спеціалізованих методів захисту цифрових мовних повідомлень, які враховують інформаційні характеристики повідомлень. На основі аналізу теоретичної стійкості шифрів запропоновано використання алгоритмів стиснення для зменшення обсягу сигналу та усунення надмірності. В якості перспективного алгоритму стиснення, заснованого на інформаційних характеристиках стисненої послідовності, запропоновано використання алгоритмів на основі оптимізованих D-перетворень, які характеризуються низькою трудомісткістю і, відповідно, високою швидкодією. На підставі огляду та аналізу блокових шифрів також було поставлено завдання розробки алгоритмів захисту голосових повідомлень за допомогою стиснення на основі оптимізованих D-перетворень другого порядку та параметризації алгоритму шифрування.

Розглянуто статистичні властивості D-послідовностей, що показують високу близькість розподілу D-бітів до рівномірного, що дає можливість використовувати більш прості алгоритми захисту в порівнянні з існуючими блочними шифрами. Проаналізовано вплив інверсії біта на поведінку та руйнування демодульованої функції. Розглянуто доцільність використання основних криптографічних операцій (підстановки, перестановки, гамування) для захисту дельта-послідовностей.

Для розробки схеми параметризації алгоритмів захисту голосових повідомлень за допомогою стиснення на основі оптимізованих D-перетворень другого порядку (алгоритм A1) пропонується використовувати алгоритм Rijndael на основі узагальнених мереж SP, затверджений як новий стандарт криптографічного захисту і дозволяє забезпечити більшу параметризацію алгоритму. Дано математичне обґрунтування роботи алгоритму, що складається з декількох шарів для забезпечення максимальних характеристик змішування та розсіювання на основі модульної арифметики. Для збільшення швидкодії

алгоритму, на відміну від традиційних схем розширення ключів, для генерації раундових ключів пропонується використовувати генератор псевдовипадкових послідовностей на основі множення з перенесенням. Виходячи із запропонованого підходу до створення раундових ключів, розглядається використання декількох режимів генерації ключів. Розглянуто можливі змінні параметри алгоритму Rijndael, запропоновано використання породжуючого многочлена, множника та вектора афінного перетворення, параметрів генератора тощо.

На основі аналізу застосування основних криптографічних операцій (підстановок, перестановок, гам) для захисту дельта-послідовностей та властивостей дельта-послідовностей розроблено симетричний блоковий алгоритм захисту стиснених голосових повідомлень (алгоритм А2) на основі оптимізованих дельта-перетворень другого порядку, що містить операції інверсії бітів, підстановки, перестановки та гамування з використанням параметризації для збільшення стійкості. Оцінюється стійкість розробленого алгоритму та розглядається вплив параметрів алгоритму на стійкість.

Отримано оцінки обчислювальної трудомісткості для відомих шифрів і запропонованих (А1 і А2) алгоритмів захисту голосових повідомлень, стиснених на основі D-перетворень другого порядку, що показують можливість значного (в 2,5-5 рази) зниження обчислювальної трудомісткості шифрування в порівнянні з існуючими алгоритмами для А1, розроблений алгоритм А2 працює приблизно в 1,5 рази швидше, ніж модифікований алгоритм А1.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Науменко М.І., Стасев Ю.В., Кузнецов О.О. Теоретичні основи та методи побудови алгебраічних блокових кодів. – Х.: ХУ ПС, 2005. – 267 с.

2. Yokoyama K., Nakagava S., Katayama H. An experimental digital videotape recorder //SMPTE J. – 1980. – 89, N 3. – P. 173.180.

3. Бернард Скляр Цифровая связь. Теоретические основы и практическое применение. Изд. 2-е, испр.: Пер.с англ.- М.: Издательский дом «Вильямс», 2003.- 1104с.

4. Oleynikov V. N , Zubkov O. V., Kartashov V. M., Korytsev I. V., Babkin S. I., Sheiko S. A. Investigation of detection and recognition efficiency of small unmanned aerial vehicles on their acoustic emission; Telecommunications and Radio Engineering, 2019, Volume 78, Issue 9; pp. 759-770.

5. Kartashov, V., Oleynikov, V., Koryttsev, I., Zubkov, O., Babkin S., Sheiko, S. Processing and Recognition of Small Unmanned Vehicles Sound Signals. 2018 International Scientific-Practical Conference on Problems of Infocommunications. Science and Technology (PIC S and T 2018) – Proceedings, 31 January 2019; pp. 392-396.

6. Kartashov V., Oleynikov V., Koryttsev I., Sheyko S., Zubkov O., Babkin S., Selieznov I. Use of Acoustic Signature for Detection, Recognition and Direction Finding of Small Unmanned Aerial Vehicles; 2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 25-29 Feb. 2020; pp. 1-4.

7. Kartashov, V.M., Oleynikov V.N, Zubkov, O.V., Korytsev I.V., Babkin, S. I., Sheiko, S.A., Kolendovskaya, M.M. Spatial-temporal Processing of acoustic Signals of Unmanned Aerial Vehicles; Telecommunications and Radio Engineering, 2020. Vol. 79, Iss, 9; pp. 769-780.

8. V. Oleynikov, O. Zubkov, V. Kartashov, I. Koryttsev, S. Sheiko, S. Babkin, "Experimental estimation of direction finding to unmanned air vehicles algorithms

efficiency by their acoustic emission"; 2019 International Scientific-Practical Conference: Problems of Infocommunications. Science and Technology (PIC S and T 2019) – Proceeding, 2019, pp. 175–178.

9. V.V. Semenets, V.M. Kartashov, V.I. Leonidov. Features of Acoustic Noise of Small Unmanned Aerial Vehicles; Telecommunications and Radio Engineering, 2020. Vol. 79, Iss. 11, pp. 985-995. DOI: 10.1615/TelecomRadEng.v79.i11.80.

10. Kartashov, V.M., Oleynikov V.N, Zubkov, O.V., Korytsev I.V., Babkin, S. I., Sheiko, S.A., Kolendovskaya, M.M. Spatial-temporal Processing of acoustic Signals of Unmanned Aerial Vehicles; Telecommunications and Radio Engineering, 2020. Vol. 79, Iss, 9; pp. 769-780.

11. Kartashov V. M., Tikhonov V. A. , Voronin V. V. Features of Construction and Application of Complex Systems for the Atmosphere Remote Sounding; Telecommunications and Radio Engineering, 2017, Volume 78, Issue 8; pp.743- 749.

12. Карташов В.М., Олейников В.Н., Колендовская М.М., Тимошенко Л.П., Капуста А.И., Рыбников Н.В. Комплексирование изображений при обнаружении беспилотных летательных аппаратов// Радиотехника. (Харьков). 2020. Вып. 201; С.120-129.

13. Карташов В.М. Модели и методы обработки сигналов систем радиоакустического и акустического зондирования атмосферы. -Харьков: ХНУРЭ, 2011. - 234 с.

14. Avalos-Gonzalez, D., Sergiyenko, O., Hernandez-Balbuena, D., Tyrsa,V., Kartashov V.M., V.,Rivas-Lopes, M., Murrieta-Rico, F.N. Constraints definition and application optimization based on geometric analysis of the frequency measurement method by pulse coincidence// Measurement: Journal of the International Measurement Confederation (USA). - 2018. –V.126. - P. 184-193.

15. Ситнік О.В., Карташов В.М., Радіотехнічні системи. Навч. Посібник. Х.: Сміт, 2009. 448 с.

16. ДСТУ OHSAS 18002:2015. Системи управління гігієною та безпекою праці. Основні принципи виконання вимог OHSAS 18001:2007 (OHSAS

18002:2008, IDT). К. : ГП «УкрНИУЦ», 2016. 21 с.

17. ДСТУ 8604:2015 Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги. URL: http://online.budstandart. com/ua/catalog/doc-page?id_doc=71028.

18. НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. URL: http://sop.zp.ua/norm_npaop _0 _ 00-7_ 15-18_01_ua.php.

19. ДБНВ.2.5-27-2006. Захисні заходи електробезпеки в електроустановках будинків і споруд. К. : Мінбуд України, 2006. 154

20. Гігієнічна класифікація праці (за показниками шкідливості і небезпеки факторів виробничого середовища від 12.08.1986 № 4137-86. - [Електронний ресурс] - Режим доступу: http://zakon4.rada.gov.ua/laws/show/v4137400-86

21. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. - [Електронний ресурс] - Режим доступу: http://mozdocs.kiev.ua/view.php?id=1972

22. ДБН В.2.5-28-2018 Природне і штучне освітлення - [Електронний ресурс] - Режим доступу: http://document.ua/prirodne-i-shtuchne-osvitlennja-nor8425.html

23. НПАОП 0.00-7.15-18 Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. URL: http://sop.zp.ua/norm_npaop _0 _ 00-7_ 15-18_01_ua.php.

24. ДБН В.1.1-7:2016 Пожежна безпека об`єктів будівництва. Загальні ви моги. URL: http://www.poliplast.ua/doc/dbn _v.1.1-7

25. Наказ Міністерства внутрішніх справ України «Про затвердження Пра вил експлуатації та типових норм належності вогнегасників». URL: https://zakon.rada.gov.ua/laws/show/z0225-18#Text.

Додаток А

(обов’язковий)

ІЛЮСТРАТИВНА ЧАСТИНА
ДОСЛІДЖЕННЯ АЛГОРИТМІВ ЗАХИСТУ АУДІО ІНФОРМАЦІЇ

назва бакалаврської кваліфікаційної роботи

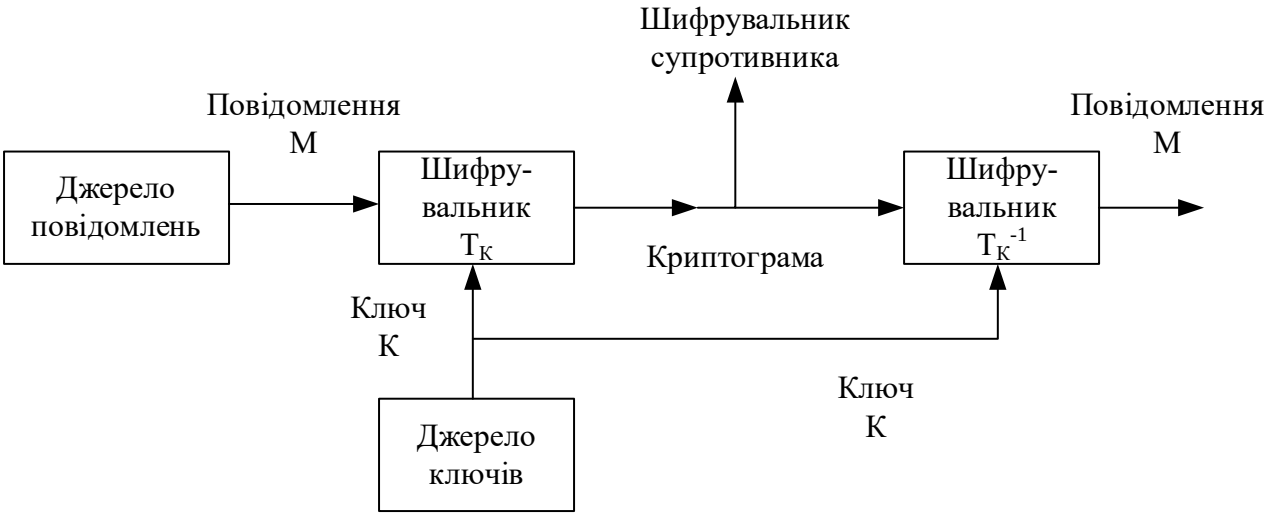
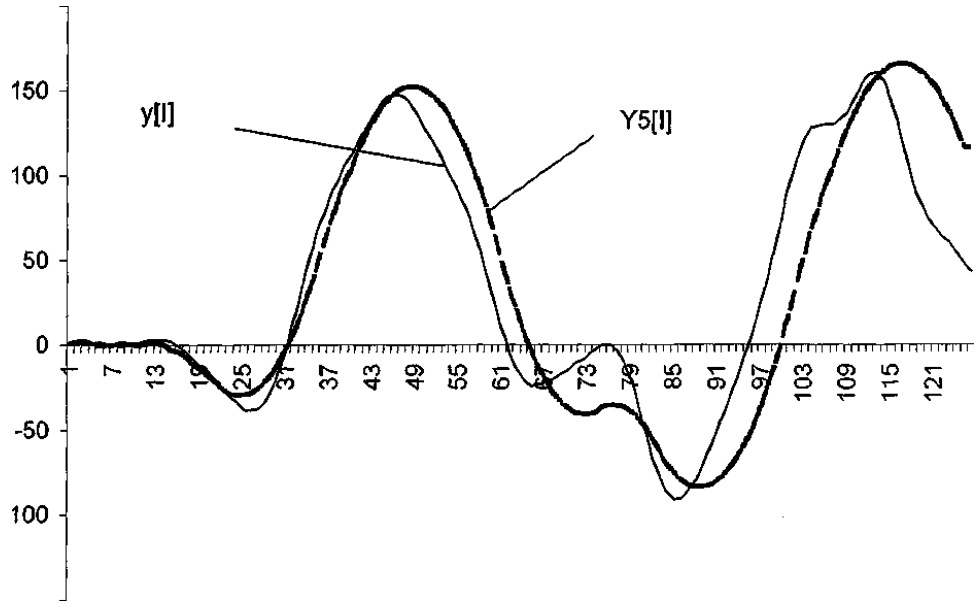
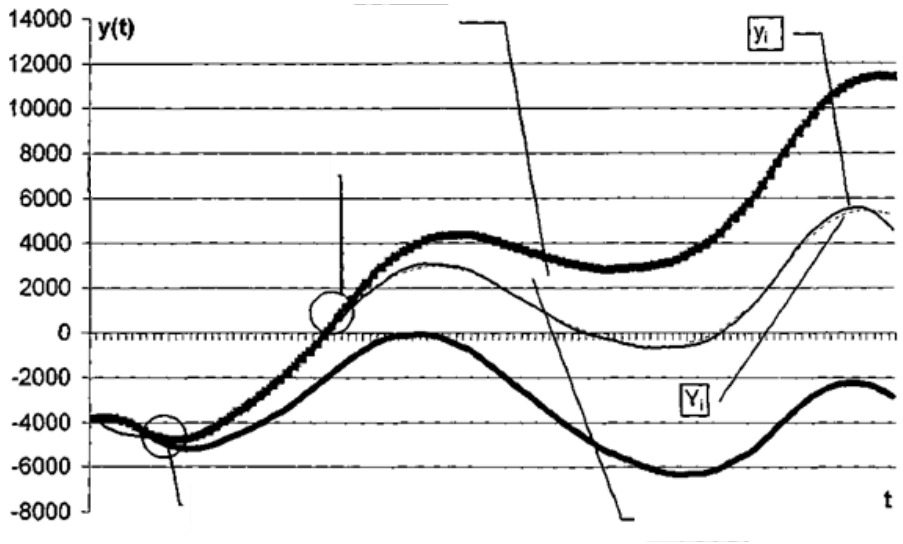


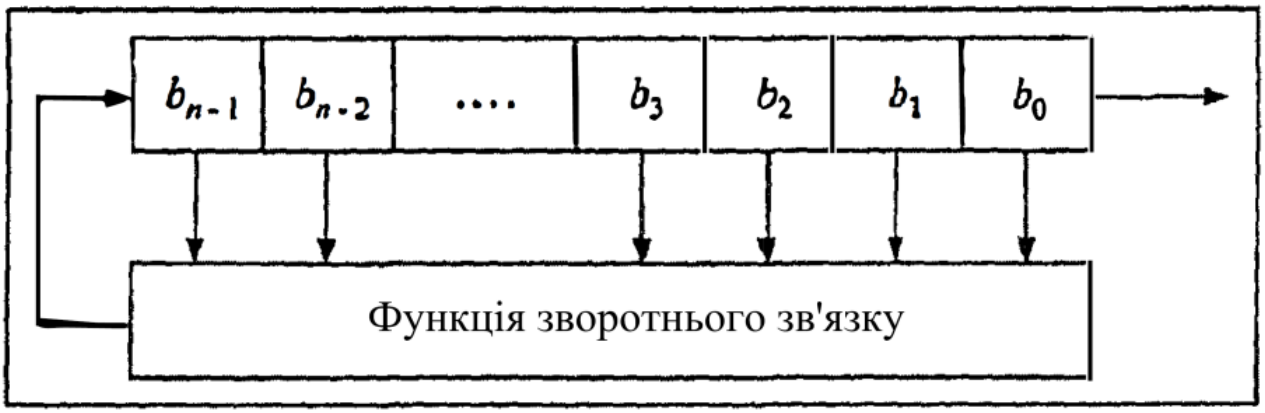
Схема системи узагальнених шифрів



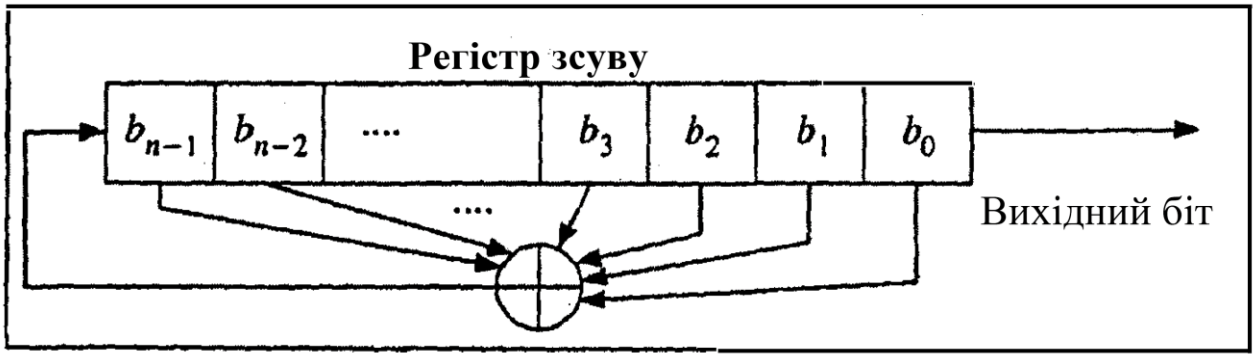
Графіки модульованої функції *y[i]* та апроксимованої функції *Y5[i]* з використанням модифікованого алгоритму дельта-перетворення другого порядку зі згладжуванням (n=5)



Вплив інверсії бітів (Y_{inv}(-) - від "+1" до "-1", Y_{inv}(+) - від "-1" до "+1") D-послідовність на демодульовану функцію для сигналу мовлення з частотою дисретизації 32 кГц, крок прогнозування n = 5.



Регістр лінійного зсуву



Регістр зсуву з лінійним зворотним зв'язком

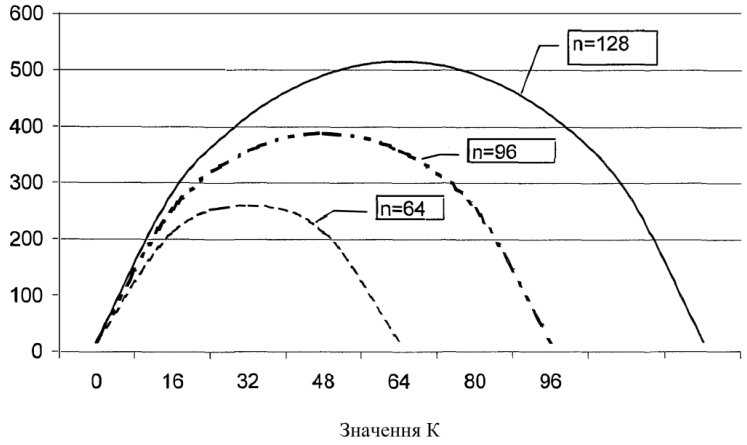


Рисунок 1 – Залежність W_{inv} для кадру перетворення довжиною n=64, 96, 128 і довжина блоку m=4.

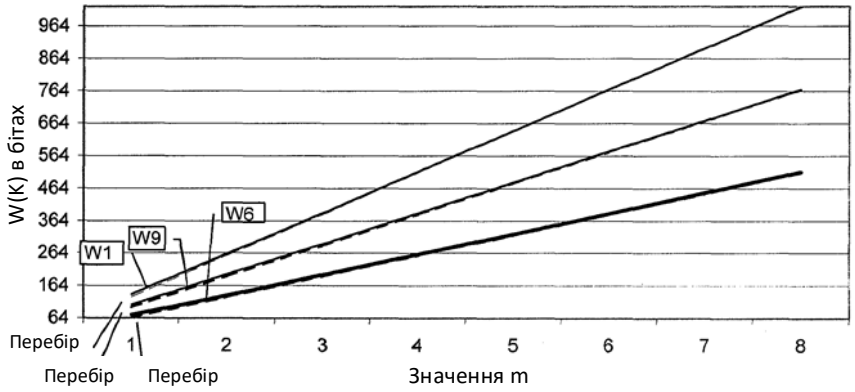


Рисунок 2 – Залежність W від m для системи перетворення довжиною n = 64, 96, 128 при $k=n/2$.

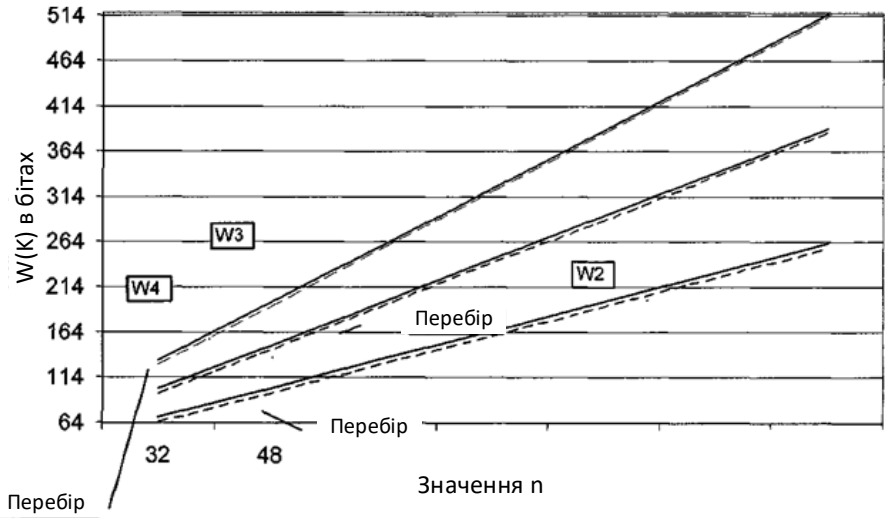


Рисунок 3 – Залежність W від n для одиниці перетворення довжини m=2,3,4 кадру при k=n/2.

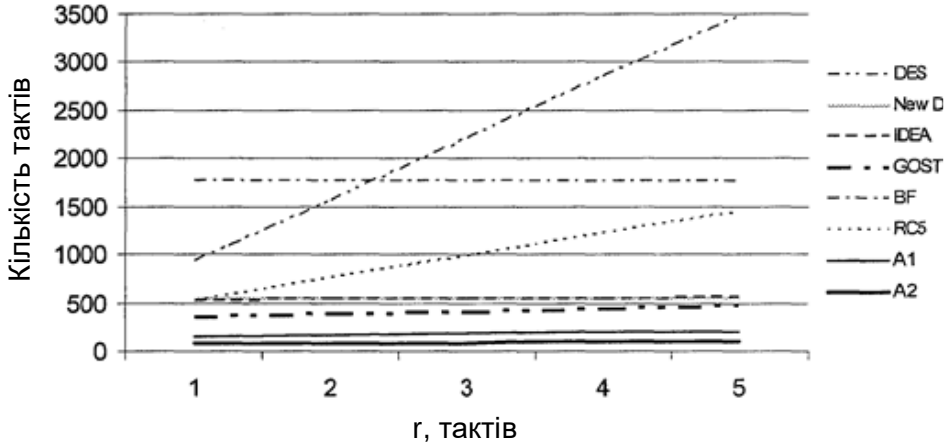


Рисунок 4 – Залежність продуктивності алгоритму від кількості тактів процесора, необхідних для виконання операції зсуву r (m=13, t=l)

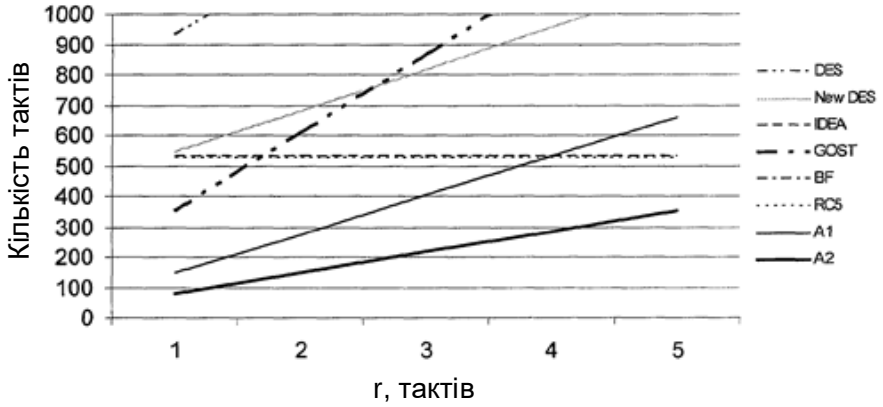


Рисунок 5 – Залежність швидкодії алгоритмів від кількості тактів процесора, необхідних для виконання операції пересилки t (m=13, r=3)

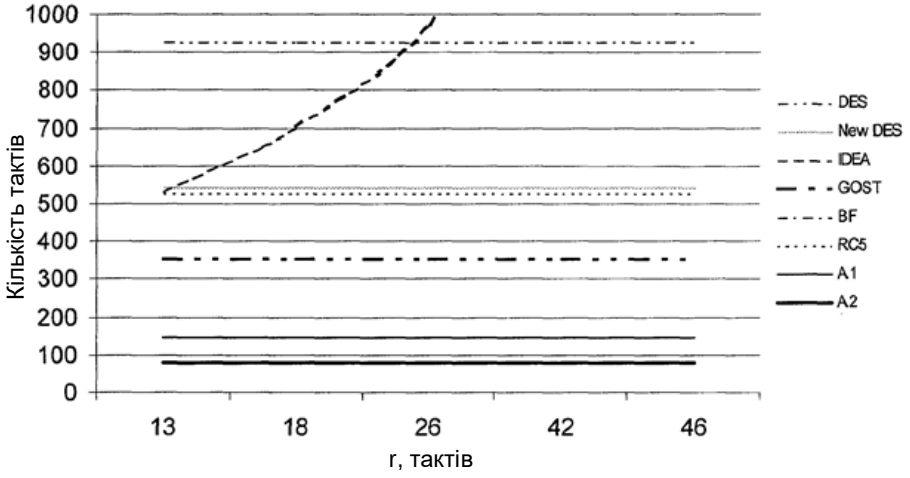


Рисунок 6 – Залежність продуктивності алгоритму від кількості циклів процесора, необхідних для виконання операції множення m (для t=l, r=3)

Додаток Б

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Дослідження алгоритмів захисту аудіо інформації

Тип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра ІКСТ, факультет ІЕС, група ПЗТ-21б
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 3,88 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

☑ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту

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

☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Кичак В.М., завідувач кафедри ІКСТ _____
(прізвище, ініціали, посада) (підпис)

Васильківський М.В., гарант ОПП _____
(прізвище, ініціали, посада) (підпис)

Особа, відповідальна за перевірку _____ Васильківський М.В.
(підпис) (прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник _____ Стальченко О.В., доцент кафедри ІКСТ
(підпис) (прізвище, ініціали, посада)

Здобувачка _____ Коновченко Д.О.
(підпис) (Прізвище, ініціали)