Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Метод та засіб шифрування на основі квазігрупових тотожностей»

Виконав: студент 2 курсу, групи 1БС-23 м
спеціальності 125 Кібербезпека та захист інформації
_____ Роман КАЧАЙ

Керівник: к. фіз.-мат. н., доцент каф. ЗІ
_____ Галина ШЕЛЕПАЛО
«17» 12 2024 р.

Опонент: к. т. н. доцент каф. ПЗ
_____ ВолодимирМАЙДАНЮК
«17» 12 2024 р.

Допущено до захисту
Завідувач кафедри ЗІ
д. т. н., проф.
_____ Володимир ЛУЖЕЦЬКИЙ
«17» 12 2024 р.

Вінниця ВНТУ – 2024 року

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Рівень вищої освіти II (магістерський)
Галузь знань – 12
Інформаційні технології
Спеціальність – 125 Кібербезпека та захист інформації
Освітньо-професійна програма – Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ,
д. т. н., проф.
_____Володимир ЛУЖЕЦЬКИЙ
«18» 09 2024 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Качаю Роману Володимировичу

1. Тема роботи: «Метод та засіб шифрування на основі квазігрупових тотожностей» керівник роботи: Шелепало Галина Василівна, к. фіз-мат. н., доцент кафедри ЗІ, затверджені наказом №310 ректора ВНТУ від 17.09.2014
2. Строк подання студентом роботи 17.12.24р.
3. Вихідні дані до роботи:
— Криптографічні перетворення;
— Список квазігрупових тотожностей;
— Генератор псевдовипадкових послідовностей;
— Набір латинських квадратів;
4. Зміст текстової частини:
Вступ. Огляд інформаційних джерел. Метод шифрування на основі квазігрупової тотожності. Засіб шифрування даних. Економічна частина. Висновки. Перелік використаних джерел. Додатки.
5. Компоненти Edon80 та їх співвідношення, квазігрупи використані в дизайні Edon80, схематичне зображення квазігрупової операції Edon80, довільний

латинський квадрат 8-го порядку, нормалізований латинський квадрат 8-
порядку, структура засобу шифрування.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завда прий
1	Шелепало Г.В., к.ф-м. н., доц. каф. ЗІ		
2	Шелепало Г.В., к.ф-м. н., доц. каф. ЗІ		
3	Шелепало Г.В., к.ф-м. н., доц. каф. ЗІ		
4	Ратушняк О.Г.,к.т.н.,доц,каф. ЕПВМ		

7. Дата видачі завдання — 1 вересня 2024року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної Роботи	Строк виконання етапів роботи	Приміт
1	Аналіз завдання. Вступ	01.09.2024 – 10.09.2024	
2	Аналіз літературних джерел за напрямком магістерської кваліфікаційної роботи	10.09.2024 – 15.09.2024	
3	Науково-технічне обґрунтування	16.09.2024 – 22.09.2024	
4	Аналіз предметної області та формування вимог до програмного засобу	23.09.2024 – 29.09.2024	
5	Розробка архітектури програмного засобу	30.09.2024 – 12.10.2024	
6	Програмна реалізація засобу	14.10.2024 – 02.11.2024	
7	Тестування розробленого засобу	03.11.2024 – 10.11.2024	
8	Розробка розділу економічного обґрунтування доцільності розробки	11.11.2024 – 17.11.2024	
9	Аналіз виконання ТЗ, висновки	18.11.2024 – 24.11.2024	
10	Оформлення пояснювальної записки	25.11.2024 – 30.11.2024	
11	Попередній захист та доопрацювання МКР	28.11.2024 – 01.12.2024	
12	Перевірка магістерської роботи на наявність плагіату	02.12.2024 – 14.12.2024	
13	Зображення МКР до захисту	13.12.2024 – 16.12.2024	
14	Захист МКР	17.12.2024 – 20.12.2024	

Студент _____ Роман КАЧАЙ

Керівник роботи _____ Галина ШЕЛЕПАЛО

АНОТАЦІЯ

УДК 681.325.5

Качай Р. Метод та засіб шифрування на основі квазігрупових тотожностей. Магістерська кваліфікаційна робота зі спеціальності 125 – Кібербезпека та захист інформації, освітня програма – Безпека інформаційних і комунікаційних систем. Вінниця: ВНТУ, 2024. 98 с. Укр. мовою. Бібліогр.: 20 назв; рис.: 9; табл.: 6.

Магістерська кваліфікаційна робота присвячена розробці методу та засобу шифрування на основі квазігрупових тотожностей. Підготовлено науково-дослідне та техніко-економічне обґрунтування доцільності розробки. У роботі здійснено аналіз існуючих криптографічних методів, обґрунтовано використання квазігрупових структур для створення алгоритмів шифрування та розшифрування, які забезпечують підвищену стійкість до сучасних атак. Розроблено математичну модель, яка враховує властивості квазігруп і їхнє застосування в криптографії. Розроблено програмний засіб, що дозволив протестувати запропонований метод та підтвердив його ефективність. Ілюстративна частина включає схеми алгоритмів, результати моделювання і демонстрацію роботи програмного забезпечення. В економічному розділі виконано аналіз витрат на реалізацію розробки та оцінено її рентабельність.

Ключові слова: шифрування, квазігрупа, криптоаналіз, латинський квадрат, криптостійкість, інформаційна безпека, захист інформації.

ABSTRACT

Kachay R. Method and means of encryption based on quasi-group identities. Master's qualification work in specialty 125 – Cybersecurity and information protection, educational program – Security of information and communication systems. Vinnytsia: VNTU, 2024. 101 p. In Ukrainian. Bibliography: 20 titles; Fig.: 9; Table: 6.

The master's qualification work is devoted to the development of a method and means of encryption based on quasi-group identities. A scientific research and technical and economic justification of the feasibility of the development has been prepared. The work analyzes existing cryptographic methods, justifies the use of quasi-group structures to create encryption and decryption algorithms that provide increased resistance to modern attacks. A mathematical model has been developed that takes into account the properties of quasi-groups and their application in cryptography. A software tool has been developed that allowed testing the proposed method and confirmed its effectiveness. The illustrative part includes algorithm schemes, modeling results, and a demonstration of the software's operation. The economic section analyzes the costs of implementing the development and assesses its profitability.

Keywords: encryption, quasigroup, cryptanalysis, Latin square, cryptoresistance, information security, information protection.

ЗМІСТ

ВСТУП

Ринок сучасності гостро потребує забезпечення захисту інформації, яка передається через різноманітні канали зв'язку у зв'язку із зростання обсягом даних та швидким обсягом інформаційних технологій. Однією із ключових галузей що спрямована на розв'язання цієї потреби є криптографія, яка надає можливості, що дозволяють шифрувати інформацію, роблячи її доступною лише для авторизованих осіб. Серед механізмів, що застосовуються в криптографії, важливе місце займають алгебричні структури, зокрема квазігрупові алгебри, які задаються тотожностями. Такі математичні об'єкти мають широкий спектр застосувань не лише у криптографії, а й у теорія кодування, комбінаториці, статистиці, теорії графів та ще в багатьох.

Використання квазігруп у криптографії бере свій початок з шифру Тритемія, де для шифрування будується таблиця Келі адитивної групи цілих чисел за модулем 26, яка застосовується для багатоалфавітного шифрування, в де кожна буква тексту шифрується окремим рядком таблиці. Пізніше розвиток цього методу, здійснений Дж. Белазо та Л. Б. Альберті, призводить до появи нових шифрів, де шифрування здійснюється для довільних алфавітів. Крім того квазігрупи та їх аналоги застосовуються у модифікаціях криптографічного алгоритму A5/1 з метою забезпечення комунікацій пристроїв інтернету речей якому використана байтова обробка інформації замість бітової, а сам вузол накладання шифру побудований на багато алфавітній заміні латинського квадрата. Використання квазігрупових тотожностей у алгоритмах шифрування дозволяє створити інноваційні методи шифрування, що забезпечують високу стійкість до можливих атак. Особливо актуальним стають в нагоді квантові квазігрупи записані тотожностями, що забезпечують захист інформації у контексті розвитку квантових технологій, тому що традиційні криптографічні алгоритми уже можуть становити загрозу для конфіденційності інформації.

Метою дослідження є підвищення рівня криптографічної стійкості та ефективності захисту інформації шляхом розробки методу та засобу

шифрування даних на основі квазігрупової тотожності.

Для досягнення мети сформульовані такі задачі:

- Проаналізувати відомі методи шифрування на основі квазігрупових тотожностей
- Розробити метод шифрування на основі квазігру, що описують многовид за законом Шредера
- Розробити структуру засобу для шифрування та проаналізувати його рівень криптографічної стійкості.

Об'єктом дослідження є процес шифрування інформації з використанням алгебричних методів та алгоритмів.

Предметом дослідження є метод шифрування на основі квазігрупових тотожностей, що описують многовиди квазігруп за Законом Шредера.

Науковою новизною є теоретичне обґрунтування вибору квазігрупової тотожності, для методу та засобу шифрування на основі закону Шредера, що дозволяє зменшити апаратні витрати.

Практична цінність полягає в отриманні нового методу та засобу для шифрування побудованого на основі квазігрупової алгебричної структури за законом Шредера.

1 ОГЛЯД ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1.1 Основні поняття та властивості квазігруп

Одним з відомих прикладів застосування квазігруп у криптографії є шифр Віженера. Цей поліалфавітний метод шифрування, описаний Джованні Беллазом, ґрунтується на послідовному використанні декількох шифрів Цезаря, кожен з яких має різні значення зсуву. Шифр Віженера використовує таблицю, відому як квадрат Віженера, що є різновидом латинського квадрата, де кожний рядок зсувається на певну кількість позицій, створюючи різні варіанти шифру Цезаря. Станом на сьогодні достатня кількість шифрів побудована на основі квазігруп, що є алгебричною конструкцією для латинських квадратів.

Латинські квадрати, невід'ємна частина вивчення квазігруп, тому що вони є їх комбінаторним аналогом. Латинські квадрати знаходять практичне застосування в різних сферах, зокрема в плануванні експерименту та теорії кодування. Латинський квадрат порядку n ─ це масив n×n, заповнений в різними символами, кожен з яких зустрічається рівно один раз у кожному рядку та рівно один раз у кожному стовпці. Таблиця Келі, введена А.Д. Келі, є значним прогресом у побудові латинських квадратів. Підхід Келі дозволяє систематично генерувати латинські квадрати за допомогою взаємно-ортогональних діагональних латинських квадратів, пропонуючи структурований метод для побудови латинських квадратів із бажаними властивостями. Ця інновація виявилася неоціненною в розробці експериментів і кодів для ви правлення помилок, демонструючи міждисциплінарну природу латинських квадратів і їх корисність за межами теоретичної математики.

Властивість латинського квадрата, проявляється в структурі таблиці Келі забезпечуючи практичний засіб забезпечення унікальної характеристики парності, властивої квазігрупам. При скінченній квазігрупі внутрішня частина таблиці Келі є латинським квадратом, тобто таблиця множення скінченної квазігрупи утворює латинський квадрат. І навпаки, довільний латинський

квадрат може бути вибраний за таблицю множення, щоб утворити квазігрупу, так само і за довільною квазігрупою можна побудувати латинський квадрат [1]. Синергія між таблицею Келі та теорією квазігруп підкреслює складні зв'язки між алгебричними структурами та їх застосуваннями в реальному світі.

Квазігруповою тотожністю називається [2] універсальна алгебра $(Q;\cdot;/;\backslash)$ сигнатури , що задовольняє такі тотожності для $\forall x,y \in Q$

$$x\cdot(x\backslash y) = y$$

$$(y/x)\cdot x = y$$

$$x\backslash(x\cdot y) = y,$$

де Q- бінарний групоїд, з визначеним на ньому набором бінарних операцій (,/,\).

В свою чергу, інші означення доповнюють дане означення та додає до опису квазігрупового рівняння наступні рівності:

$$(y\cdot x)/x = y$$

$$x/(y\backslash x) = y$$

$$(x/y)\backslash x = y$$

З цих властивостей випливає, що операціям на квазігрупах властиве поняття асоціативності, що відрізняє їх від інших алгебричних структур, таких як групи, де операція повинна бути асоціативною.

Квазігрупа, яка має двосторонній одиничний елемент, називається лупою. Лупа ─ це квазігруна з додатковою властивістю, для якої існує такий елемент e, що виконується така тотожність:

$$a\cdot e = e\cdot a = a \text{ для } \forall a \in Q.$$

Квазігрупа називається тотально-симетричною, якщо всі її оборотні операції збігаються, тобто латинський квадрат один для многовиду таких квазігруп. Оборотною називається операція, якщо для неї існує обернений елемент.

1.2 Означення та властивості квазігрупових тотожностей

Ідентифікація квазігрупових тотожностей мінімальної довжини має особливе значення. Коротші тотожності не тільки забезпечують більш стисле зображення визначальних властивостей квазігрупи, але також пропонують розуміння основних компонентів, які відрізняють один клас квазігруп від іншого. Часто виникає питання про створення квазігруп з визначеними властивостями особливо це стосується шифрування.

Оптимальними методами для такого створення є алгебричні методи, які передбачають попереднє вивчення конкретних класів квазігруп. Найбільш вивченими серед усіх таких класів є многовиди, тобто класи квазігруп, що визначаються за допомогою тотожностей.

Кажуть, що тотожність має тип (n; m), якщо вона має дві предметних змінних, кожна з яких має відповідну кількість поява, а саме перша змінна ─ n поява, друга змінна ─ m поява.

У дисертації Г.В. Шелепало [1] вказано, що кожна скінченна квазігрупа $(Q;\cdot)$ задовольняє деяку тотожність типу (m;2) для деякого m. З цього твердження випливає доцільність вивчення многовидів квазігруп, що визначаються тотожностями типу (m;2). В.Д. Білоусов [1] виділив множини тотожностей квазігруп типу (3;2), в яких певні парастрофи взаємно-ортогональні. Більш точно, досліджено, що відповідно до парастрофної еквівалентності таких класів існує сім варіантів, які він продемонстрував у вигляді семи тотожностей. Дані тотожності типу (3;2) визначають особливості та взаємозв'язки в цих квазігрупах.

Дві тотожності вважаються парастрофно рівносильними, якщо одну можна отримати з іншої за допомогою обмеженої кількості застосувань рівностей:

$$(x\backslash y)\cdot y = x, \qquad x\cdot(x/y) = y$$

та перейменуванням функційних і предметних змінних.

Проте, як виявилося, навіть уже квазігрупові тотожності типу (2;2) мають широке застосування в криптографії. А саме, шифр Edon L будується з використанням квазігрупової тотожності типу (2;2), тобто

$$xy \cdot y = x.$$

Крім цього, інша квазігрупова тотожніть типу (2;2) використовується для методу шифрування Edon R, зокрема

$$x \cdot xy = y.$$

Мінімальні квазігрупові тотожності типу (3;2) теж мають влучне застосування в шифруванні. А саме, перший закон Білоусова, на якому описаний алгоритм шифру Edon 80, задовольняє квазігрупову тотожність

$$x \cdot (x \cdot xy) = y.$$

Тотожність першого закону Білоусова лягла в основу багатьох різноманітних криптосистем [10-13,15].

Після ознайомлення з основними поняттями та твердженнями математичного апарату квазігрупових тотожностей можна провести аналіз існуючих криптосистем на їх основі.

1.3 Аналіз існуючих криптосистем

1.3.1 Коди автентифікації повідомлень

Код автентифікації повідомлень (MAC) є інструментом для забезпечення цілісності даних та автентифікації джерела.

Алгоритми коду автентифікації повідомлень (MAC) використовуються для перевірки цілісності даних та автентичності джерела. Вони отримують

повідомлення різного розміру та секретний ключ і на основі цих даних генерують спеціальний тег або MAC-значення. Отримувач, який має той самий секретний ключ, може за допомогою цього тегу впевнитись, що дані не були змінені і що вони походять з надійного джерела. Існують різні способи створення MAC на основі блокових шифрів або хеш-функцій.

Хоча в літературі можна знайти кілька квазігрупових схем MAC [3], тут опишемо QMAC [4]. Меєр [4] побудував QMAC – алгоритм MAC, який базується на загальновідомій квазігрупі (Q,*). Для підвищення рівня безпеки рекомендується використовувати велику, дуже неасоціативну квазігрупу без чіткої структури. Секретний ключ K довжини t визначається як схема дужок для елементів m...m, (тобто вибір дужок для m * *m), разом з константою c, яка належить Q. Ця константа c необхідна для захисту від адаптивних атак, які можуть виникати шляхом модифікації кожного внутрішнього множення $(m * m_{i+1})$ на $((m * c) * m_{i+1})$. Таким чином, тег для повідомлення $M = m...m_i$, де m_i належить Q та i знаходиться в межах {1...1}, позначений як $h_k(M)$, обчислюється через квазігрупову операцію * за певним порядком для ключа K з урахуванням описаного правила про константу c для кожного внутрішнього виконання операції *.

Нехай (Q. *) — квазігрупа і M = 0223103. Нехай K довжини 7 складається зі схеми круглих дужок, яка і визначає своєрідну квазігрупову тотожність:

$$((m_1 * m_2) * m_3) * (((m_4 * m_5) * m_6) * m_7)$$

і c = 1. Маємо:

$$h_K(M) = (((0 * 1) * 2) * 2) * ((((3 * 1) * 1) * 0) * 3)$$
$$= ((1 * 2) * 2) * (((3 * 1) * 0) * 3)$$
$$= (1 * 2) * ((3 * 0) * 3)$$
$$= 1 * (0 * 3)$$
$$= 1 * 3$$
$$= 2.$$

Є три різні методи, як обчислити значення MAC для довших повідомлень, розбиваючи повідомлення M на блоки, кожен з елементами Q. Якщо останній

блок менший за 1, застосовується доповнення. Нехай M M_1... M_y з доповненням. Для одного з методів побудова значення MAC задається такими рівняннями:

$$H_o = IV \in Q$$
$$H_i = H_{i-1} * h_K (M_1),\ 1 \le i \le N$$
$$QMAC_K(M) = H_N$$

Присутнє зображення ключа довжини 1 з перестановкою 1-1 елементів, і при цьому обмежує розмір ключового простору (t-1)! Q, фактичний розмір набагато менший, тому що різні перестановки можуть входити в ту саму схему дужок. Тим не менш, розмір ключового простору експоненціально збільшується з довжиною ключа. Наприклад, для розміру ключового простору 2128 і |Q| = 28, потрібно $1 \le 66$.

1.3.2 Потоковий шифр Edon80

Edon80 — це двійковий адитивний потоковий шифр, який виявився незламаним і став фіналістом конкурсу eSTREAM [5]. Edon80 працює в трьох основних режимах: налаштування ключа (KeySetup), налаштування вектора ініціалізації (IVSetup) і режим генерації ключового потоку (Keystream mode). Для коректної роботи Edon80, окрім його основної частини (ядра), необхідні такі додаткові ресурси:

1. Регістровий ключ обсягом 80 біт для зберігання фактичного секретного ключа,

2. Регістр IV обсягом 80 біт для зберігання вектора ініціалізації з доповненням,

3. Внутрішній 2-бітовий лічильник, який використовується як фідер для ядра Edon80 у режимі генерації ключового потоку (Keystream mode).

Алгоритм розшифрування

Вхід: елементи сектретного ключа K = k_1k_2... k_{128}, де k_i — біти,

зашифроване повідомлення $C_1C_2... C_{64}$, C_i — біти.

Вихід: текстове повідомлення a = a_1a_2….. a_{64}.

Ініціалізація: поставте (k_{2i-1}, k_{2i}) для i = 1, 2, ..., 8.

Пошук квазігрупи Q; використовуючи двійковий порядковий номер, представлений як (k_{7i-6}, k_{71-5}. k_{7i}), де i = 1, 2, ..., 8. Ініціалізуйте матриці AQ і Bo, а також вектор mo, для i = 1 ,..., 8.

Знайти квазігрупу T_i використовуючи двійковий порядковий номер, поданий як ($k_{7(i+8)-6}$, $k_{7(i+8)-5}$..... $k_{7(i+8))}$. Ініціалізувати матриці $A_т$ і $B_т$, а також вектор $m_т$ для i = 1, 2, ..., 8.

Для i =1 до 64 $a_i \leftarrow$ c

Для i = до 4

ltmp ← l_{i+4};

для j = 64 до 2 крок 2

tmp ←(a_{j-1}.a_j);

(a_{j-1}, a_j) = B−1 T_{i+4} mT T_{i+4} + B−1 T_{i+4} (I + C)AT_{i+4} l T tmp+ B−1 T_{i+4} (CmT T_{i+4} ◦ CAT_{i+4} l T tmp) + B−1 T_{i+4} tmpT + B−1 T_{i+4} (CAT_{i+4} l T tmp ◦ CtmpT);

hmp (a_{j-1}, a=);

ltmp ← l_i;

для j 1 до 63 крок 2

tmp ← (a_j. a_{j+1});

(a_{j-1}, a_j) = $B^{-1}{}_{Ti}$ $m^{T}{}_{Ti}$ + $B^{-1}{}_{Ti}$ (I + C)AT_i T tmp+ B−1 T_i (CmT Ti ◦ CATi l T tmp) + B−1 Ti tmpT + B−1 T_i (CATi l T tmp ◦ CtmpT);

ltmp ← (a_j , a=);

для i = 1 до 8

ltmp ← l_i;

для j = 1 до 7 крок 2

tmp ← (a_j , a_{j+1});

(a_{j-1}, a_j) = B−1 Qi mT Qi + B−1 Qi (I + C)AQi l T tmp+ B−1 Qi (CmT Qi ◦ CAQi l T tmp) + B−1 Qi tmpT + B−1 Qi (CAQi l T tmp ◦ CtmpT);

ltmp ← (a_j , a_{j+1});

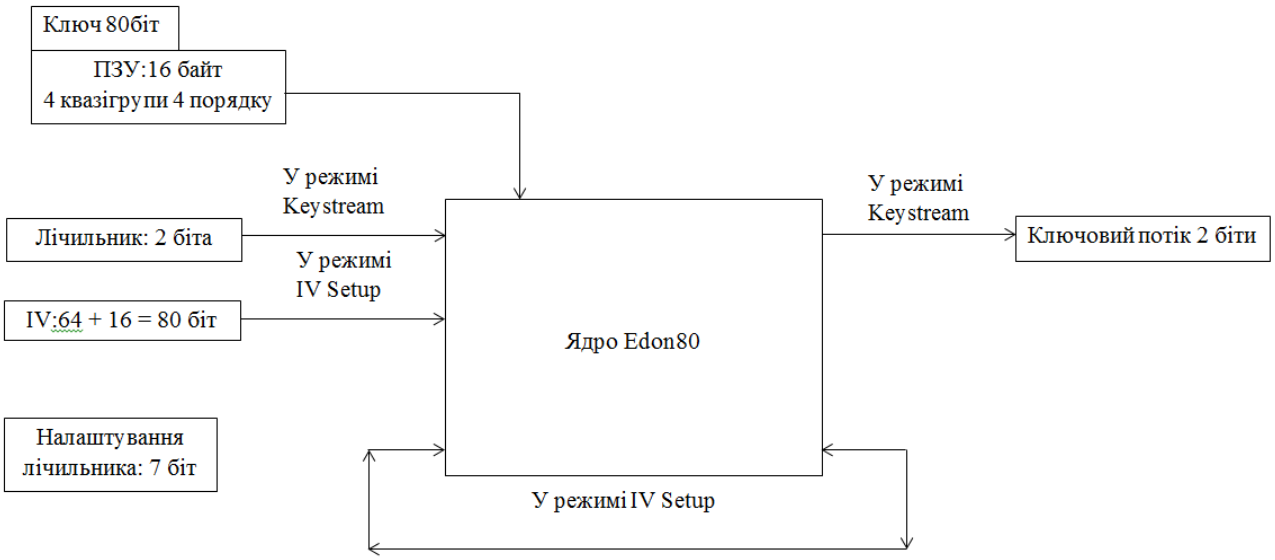


Рисунок 1.1 - Компоненти Edon80 та їх співвідношення

$\bullet_0$	0	1	2	3
0	0	2	1	3
1	2	1	3	0
2	1	3	0	2
3	3	0	2	1

$\bullet_1$	0	1	2	3
0	1	3	0	2
1	0	1	2	3
2	2	0	3	1
3	3	2	1	0

$\bullet_2$	0	1	2	3
0	2	1	0	3
1	1	2	3	0
2	3	0	2	1
3	0	3	1	2

$\bullet_3$	0	1	2	3
0	3	2	1	0
1	1	0	3	2
2	0	3	2	1
3	2	1	0	3

Рисунок 1.2 - Квазігрупи, використані в дизайні Edon80

Один 7-бітний SetupCounter, який використовується в режимі IVSetup, 5. Один 4 x 4 = 16-байтовий банк ПЗУ, де містяться 4 квазігрупи (тобто латинські квадрати) порядку 4, зберігається. Ці 4 попередньо визначені квазігрупи описані на рис. 1.2.

Внутрішню структуру Edon80 можна розглядати як конвеєрну архітектуру з 80 простих 2-розрядних трансформаторів, які називаються електронними трансформаторами.

Структура, що виконує операцію * в електронних трансформаторах є квазігруповою операцією порядку 4. Тут електронний трансформатор є

квазігруповою операцією *. Отже, в Edon80 є 80 електронних трансформаторів, каскадованих у конвеєрі, де один живить інший.

Хочеться підкреслити, що обрані квазігрупи мають достатньо великі періоди зростання. Таким чином, якщо будь-яка з квазігруп використовується k разів в e-перетвореннях, то період отриманого рядка становитиме відповідно 2,66*, 2,48%, 2,43k, 2,37k. (Звернемо увагу, що 2,4880 ≈ 2104,8). Можна констатувати, що 64 з 576 квазігруп порядку 4 мають настільки великі періоди зростання, що будь-які 4 з них можуть бути взяті для побудови Edon80.

Edon80 показує, що при проектуванні квазігрупи дуже малі замовлення може виробляти криптопримітиви високої якості.

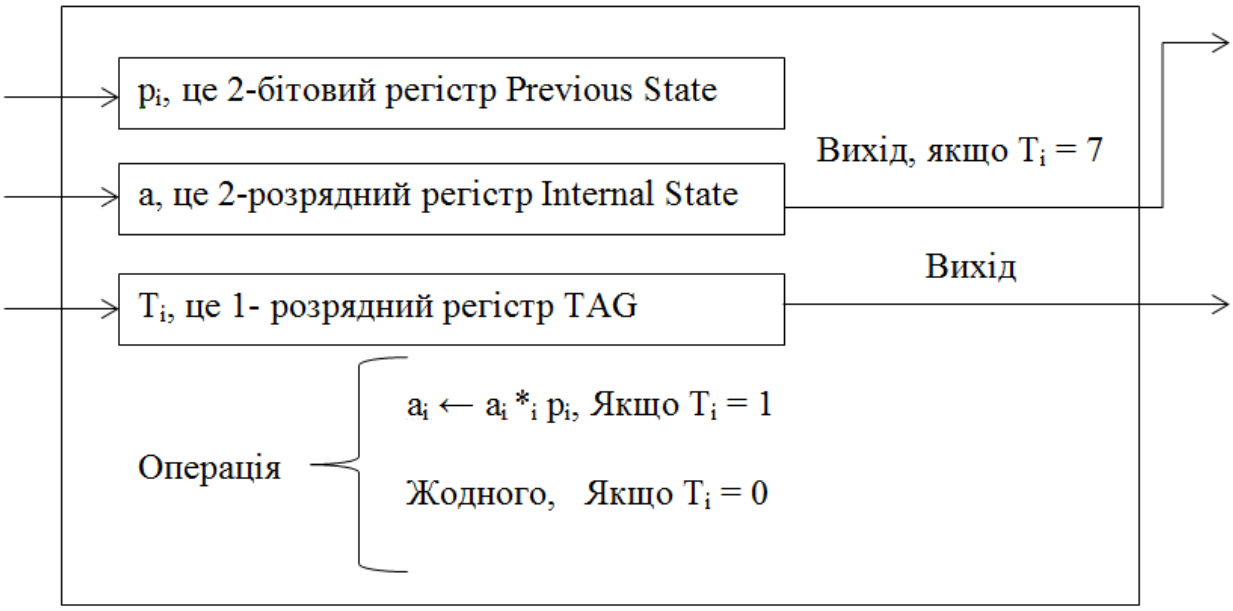


Рисунок 1.3 - Схематичне зображення квазігрупової операції Edon80

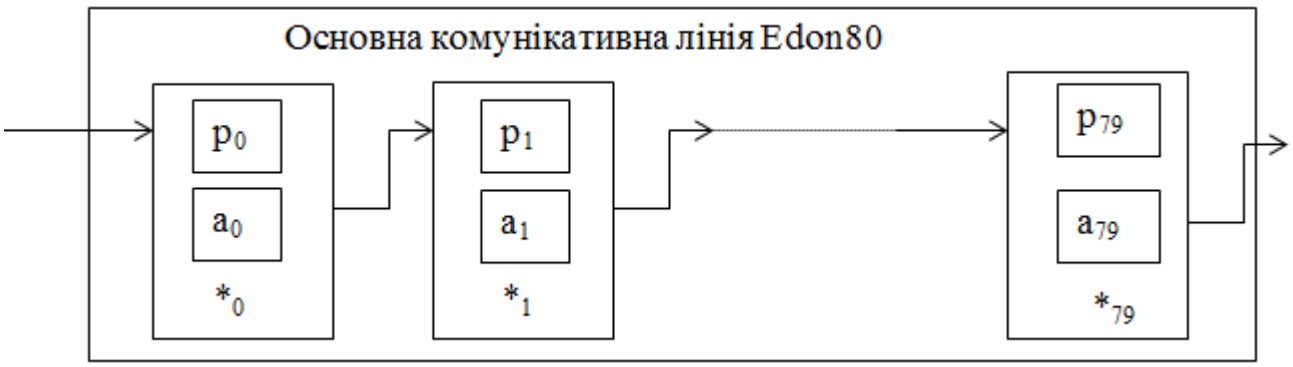


Рисунок 1.4 - Ядро Edon80 з 80 конвеєрними електронними перетвореннями

1.3.3 Автентифіковане шифрування

Автентифіковане шифрування поєднує шифрування і автентифікацію, забезпечуючи одночасно конфіденційність і цілісність даних. Цей метод дозволяє не лише зашифрувати дані, а й перевірити, чи були вони змінені під час передачі. Існує варіант автентифікованого шифрування з пов'язаними даними (AEAD), який забезпечує додаткову автентифікацію незашифрованих даних.

Одним із прикладів такого шифру є π-Cipher, який був розроблений для великих обсягів даних і має кілька варіантів реалізації, залежно від розміру ключа і цілей шифрування. π-Cipher використовує операції додавання, обертання і XOR для досягнення високої складності шифрування.

π-Cipher — це новий автентифікований шифр, який пропонується в чотирьох варіантах, залежно від цілей: 16-Cipher096 (легша версія), π32-Cipher128, π64-Cipher128 і π64-Cipher256. Цей шифр є одним із кандидатів, що бере участь у другому етапі конкурсу CAESAR з автентифікованого шифрування [6]. Він має онлайн-дизайн, підтримує паралельну та інкрементну обробку, є криптографічно захищеним і не використовує пов’язаних даних. π-Cipher базується на конструкції дуплексної губки [7], де основною частиною є функція перестановки, побудована на основі операцій ARX. У цьому підході, замість звичних S-блоків для забезпечення нелінійності та дифузії даних, використовуються операції додавання, обертання та XOR. При правильному поєднанні ці операції утворюють складну нелінійну взаємодію. В цьому огляді я коротко пояснюю π-шифр на рівні, достатньому для розуміння його основних елементів, і пропоную звернутися до для глибшої і формальної специфікації.

Загальна функція перестановки складається з трьох основних перетворень

$$\mu.\nu.\sigma: : Z4/2\ \omega \rightarrow Z4/2\ \omega\ ,$$

де Z2ω є набором усіх цілих чисел від 0 до 20. Ці перетворення виконують

роботу дифузії та нелінійного змішування вхідних даних.

Застосовуються такі операції:

- Додавання + модуль 2";
- Операція повороту ліворуч (круговий зсув ліворуч), ROTLp(X), де X — ω – розрядне число, а p — ціле число з $0 \leq p < \omega$;
- Порозрядна операція XOR для $\oplus$ на ω - розрядних чисел.

Нехай:

$$X (X_0, X_1, X_2, X_3),$$

$$Y = (Yo, Y_1, Y_2, Y_3),$$

$$Z (Zo, Z_1, Z_2, Z_3)$$

три 4-кортежі ω -розрядних слів.Далі позначимо операцію:

$$Z = X * Y = \sigma(\mu(X) \oplus {}_4 v(Y))$$

де 4 є покомпонентним додаванням двох 4-вимірних векторів.

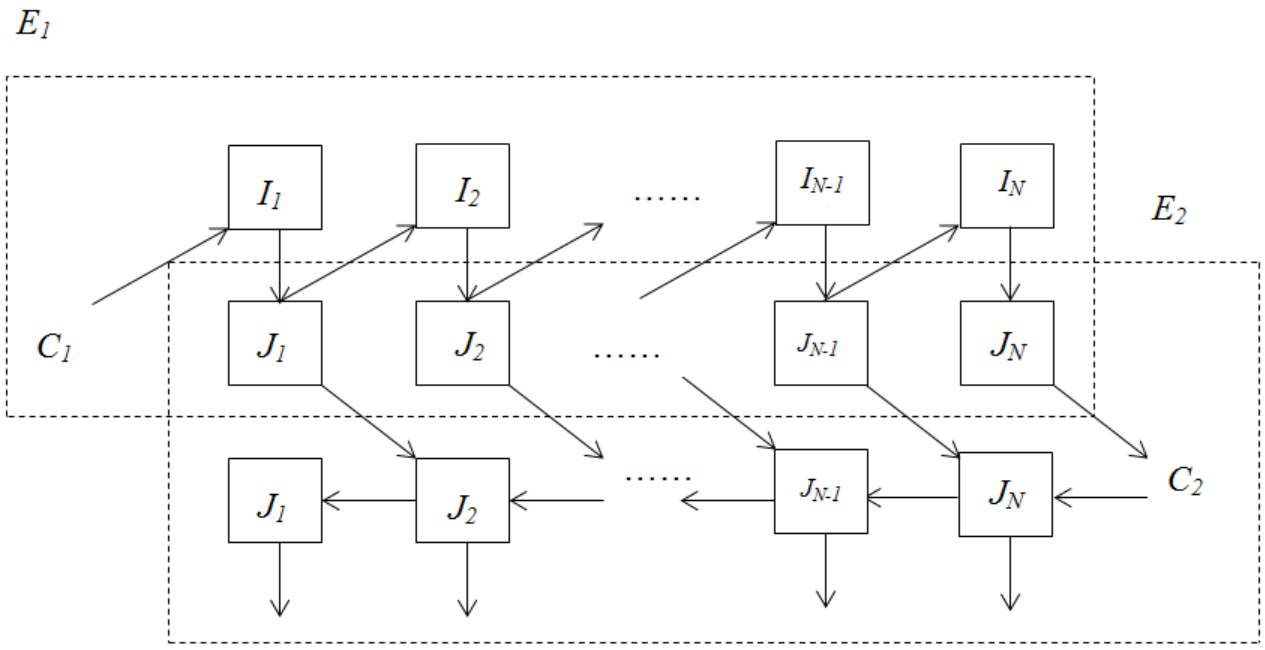


Рисунок 1.5 - Графічне зображення одного раунду 7-функції, де C∈ Qq є вектором N q-бітових постійних значень 64, 128 або 256 і 1 (11..... IN), J (J.. JN) EQ є входом і виходом функції

Операція *, є квазігруповою операцією (Qq, *), де Qq = {0, 1} є набором елементів 0 і 1 з потужністю 29 і q = 64, 128, 256. Через те, що використовуються квазігрупи порядку 29, також квазігрупові перетворення рядків згадуються у визначенні -функції. Стан функції представлено у вигляді списку з N 4-х кортежів, кожна з яких має довжину ω- бітів (bNx 4 x w).

Наступним описом автентифікованого шифрування є алгоритм AEAD-InGAGE. InGAGE — це сімейство автентифікованих шифрів із підтримкою пов'язаних даних, побудоване на основі легкої криптографічної хеш-функції GAGE. Принцип роботи InGAGE схожий на режим автентифікованих шифрів Ascon, проте між ними є кілька відмінностей, як у самому режимі, так і в використовуваних параметрах. Основним компонентом процесів шифрування та розшифрування в InGAGE є кругова перестановка під назвою QPERMUTATION, яка функціонує так само, як і хеш-функція GAGE, але з різною кількістю раундів. Полегшена хеш-функція GAGE та алгоритм автентифікованого шифрування InGAGE [8] були відібрані до першого раунду процесу стандартизації, організованого NIST, для полегшених криптографічних алгоритмів (AEAD та хеш-алгоритмів) [9].

1.3.4 Криптографія з відкритим ключем

У криптографії з відкритим ключем (PKC) для кожного учасника комунікації генерується пара відкритого та закритого ключів. Відкритий ключ, як випливає з його назви, є відкритим для всіх, і якщо хтось хоче надіслати секретне повідомлення його власнику, він повинен зашифрувати повідомлення цим відкритим ключем. Власник за допомогою свого закритого ключа, відомого тільки йому, розшифровує повідомлення. Отже, кожна схема шифрування з відкритим ключем складається з: алгоритму генерації ключа, алгоритму шифрування та алгоритму розшифрування. Алгоритми з відкритим ключем зазвичай набагато повільніші, ніж симетричні алгоритми, тому їх

основне застосування полягає в обміні секретним ключем між сторонами, що спілкуються, тоді як фактичне спілкування шифрується за допомогою деякої симетричної криптосистеми. Алгоритми відкритих ключів також використовуються в цифрових підписах, узгодженні ключів з автентифікацією пароля, протоколах невідмовності, депонуванні ключів тощо. Перша спроба розробити схему відкритого ключа на основі квазігрупи зроблена Кедвеллом, використовуючи перехрещені інверсні квазігрупи.

Багатовимірний РКС базується на NP-складній задачі розв'язування багатовимірних квадратичних (MQ) систем рівнянь над скінченними полями. Для всіх схем MQ загальним є те, що відкритий ключ будується як P = SOFT, де F є деяким легко оборотним квадратичним відображенням, тоді як S і T є бієктивними афінними перетвореннями. Одне з важливих застосувань квазігруп (зокрема багатовимірних квадратичних квазігруп), яке зацікавило криптоспільноту десять років тому, належить багатовимірному РКС. Причинами були пошуки нових постквантових рішень, і на той час схема підпису MQQ-SIG була найшвидшою схемою в тестуванні криптографічних систем ECRYPT (eBACS). Проте Faugère та ін. показав, що алгоритми з відкритим ключем на основі MQQ мають спільну алгебраїчну структуру, яка може бути використана для запуску успішного поліноміального відновлення ключем від нападу. Проте далі описано криптосистеми MQQ і MQQ-ENC, а також схему підпису MQQ-SIG.

Криптосистема MQQ була запропонована в 2008 році рядом авторів Глікороскі та ін. [10] з модифікацією квазігруп. Багатовимірні квадратичні квазігрупи використовуються для побудови багатовимірних квадратичних поліномів над скінченними полями, які використовуються для побудови алгоритму.

Опис криптосистеми пропонує також евристичний алгоритм для пошуку MQQ типу Quadd-kLink порядку не більше 2^5. Спочатку генеруються два набори з MQQ типу Quadzin і Quads Ling (кожен набір містить щонайменше мільйон елементів). Потім дві квазігрупи (*1, *2) вибираються випадковим

чином з першого набору та шість із другого набору (*3..... *8).

Далі система

$$F = (F(x_1,...,x_n), Fn\ (x_1,..., X_\eta))$$

квадратичних поліномів над F2, де n = 5k, k≥ 28, генерується за допомогою таких кроків:

1. Випадково згенерувати n булевих функцій

$$F = (f_1 \ldots. f_n) \text{ від n змінних } x = (x_1,..., x_n)$$

   і представити вектор F(x) у вигляді рядка $x_1... x_i$, де x_i — вектори розмірності 5.
2. Визначити (k-1)-кортеж

$$I = (i1,..., ik\text{-}1),\ ij \in \{1, 2, , 8),$$

   як набір індексів, щоб визначити, яка квазігрупа буде використана в нелінійному перетворенні та з вимогою, щоб індекси 1 і 2 повторювалися 8 разів у l.
3. Обчислити у Y... Y_k, де:

$$Y = X_1, Yj+1 = Xj\ *i,\ Xj+1, \text{ для } j = 1, 2.....k - 1.$$

4. Задати 13-вимірний вектор

$$Z = Y1 \parallel Yr1.1 \parallel Yr2.1 \parallel \ldots\ldots \parallel Yrg.1,$$

   де Yrj.1 означає першу координату вектора Yrj. Перетворення Z біекцією Доббертена: W = Dob(Z).
5. Установити координату:

Y_1 = (W1, W2, W3, W4, W5), Yr1.1 = W6, Yr2.1 = W7, ….Yrg.1 = W13.

6. Повернути координату у як n багатовимірних квадратичних поліномів

$$F_i(x_1,...,x_n),\ i = 1, 2, ... n.$$

де F — бієктивне багатовимірне квадратичне відображення S і T є двома неособливими лінійними перетвореннями, вибраними рівномірним випадковим чином із множини F_2. Приватний ключ 2 — це кортеж із 10 (S, T, *1.. *8), а карта P = SOFT — це відкритий ключ. Розмір відкритого ключа становить

$$n\cdot(1+(n+1)/2)$$

біт. Шифрування повідомлення m = (m...,m) ∈ F_2 здійснюється шляхом обчислення:

$$c = P(m),$$

поки розшифровка виконується:

$$m = P^{-1}(c) = T^{-1} \circ T^{-1} \circ S^{-1}(c)$$

Схема шифрування MQQ-ENC побудована на схемі шифрування MQQ з кількома відмінностями . Він розроблений як імовірнісна схема шифрування з помилками розшифрування, побудована над будь-яким малим полем Fpk (k≥ 1 і p a просте число). Функція люка використовує мінус-модифікатор із фіксованою кількістю видалених рівнянь, що руйнує бієктивність F, а в процесі розшифрування універсальна хеш-функція допомагає зробити ймовірність помилкового розшифрування незначною. Іншою важливою відмінністю є

розгортання лівих MQQs (LMQQ). Крім того, перетворення S і Tare побудовані за допомогою комбінації двох циркулянтних матриць.

Схему підпису MQQ-SIG [11] можна розглядати як (1/2) усікання багатовимірної квадратичної системи SoFoT: (0,1)" (0, 1)". Це означає, що n/2 рівнянь видаляються у відкритому ключі, що фактично є застосуванням мінусового модифікатора. Як квазігрупа використовується MQQ порядку 2^8, отримана методом з (з розміром пам'яті 81B).

Афінні перетворення S і T визначаються циркулянтними матрицями таким чином, де Iσ0, i ∈ [0,1,2,.... , n/16) і Iσ1. i ∈ (0,1,2,... n/16 + 1} є n x n матрицями перестановок, тоді як σ0 і σ1 є перестановками на n елементах, які створюють неособливу матрицю S^{-1}. Для останніх σ0 і ol є випадковою перестановкою на {1, 2, ..., n}, тоді як

$$\sigma k = \text{Rotate Left}(\sigma k\ 1, 8),\ k = \{0, 1.\ \},\ i = \{1,...n/16+k).$$

Перетворення $S = (S^{-1})^{-1}$, а T отримано з S за формулою:

$$T=S\ *x+v.$$

Схема MQQ-SIG працює наступним чином. Для генерації пари відкритого та закритого ключів спочатку генерується MQQ * разом із S і T. Потім центральне біективне багатовимірне квадратичне відображення

$$F = (F1\ (x_1,...,x_n),\ F(x_1,\ .,x)),$$

де n = 32 x k, k ∈ (5,6,7,8), генерується за такою процедурою:

1. Випадково згенерувати n булевих функцій від n змінних. Представте заданий вектор x = (f_i..... f_n) у вигляді рядка x = X_1... X, де X_i — вектори розмірності 8.

2. Обчислити у Y... Y, де: Y = X1, $Y_{j+1} = X_j * X_{j+1}$, для парних j = 2, 4..... та $Y_{j+1} = X_{j+1}*X_j$, для непарних j = 3, 5, ...
3. Повернути у як n багатовимірних квадратичних поліномів $F(x_1,...,x_n)$, i = 1, 2, ... n.

Далі обчислити y = SoFoT(x). Відкритим ключем є у як n/2 багатовимірні квадратичні поліноми $Fi(x_1,...,x)$, i = 1+n/2,...n.n, а закритим ключем є кортеж (σ0.σ1.*).

Щоб зменшити розмір відкритого ключа, розробники вирішили розділити його хешуйте результат повідомлення M на дві частини та підпишіть його, використовуючи двічі те саме повторення.

1.3.5 Блокові шифри

В методі блокового шифрування дані поділяються на блоки фіксованої довжини, і кожен блок шифрується окремо за допомогою секретного ключа. Шифрування відбувається через застосування спеціальної функції, що приймає вхідні дані та ключ, після чого перетворює їх на зашифрований блок. Для розшифрування використовується обернена функція. Важливо, що для одного і того ж самого ключа кожен блок шифрується однаково, що робить цей метод зручним для передачі великих обсягів даних [13].

Кращою властивість блокових шифрів є здатність бути перестановкою для фіксованих ключів. Блокові шифри на основі квазігруп використовуються нечасто, наприклад існує конструкція блокового шифру, де застосовується квазігрупа, зображена в матричному вигляді. Зокрема, конструкція BCDMPQ використовує матричне зображення квазігруп порядку 4. Нехай, задано квазігрупу (Q; *) порядку 4, для всіх x,y ∈ Q, $x = (x_1,x_2)$, $y = (y_1,y_2)$, x_i,y_i є бітами:

$$x * y = m^T + Ax^T + By^T + CAx^T \circ CBy^T$$

де

$$A = \begin{bmatrix} a_{11}\ a_{12} \\ a_{21}\ a_{22} \end{bmatrix} B = \begin{bmatrix} b_{11}\ b_{12} \\ b_{21}\ b_{22} \end{bmatrix} C = \begin{bmatrix} 1\ 1 \\ 1\ 1 \end{bmatrix}$$

неособливі Булеві матриці, m = [m_1, m_2] є Булевим вектором. Операція " ◦ " означає покомпонентний добуток двох векторів.

Всього існує 144 квазігрупи матричного виду. З них вибирається список із 128 і зберігається в пам'яті таким чином:

seq_num $m_1, m_2, a_{11}, a_{12}, a_{21}, a_{22}, b_{11}, b_{12}, b_{21}, b_{22}$

де seq_num є семибітним числом (точніше номер квазігрупи в списку), тоді як m_1, m_2, a_{11}, a_{12}, a_{21}, a_{22}, b_{11}, b_{12}, b_{21}, b_{22} є бітами, що з'являються в матричній формі квазігрупової операції. (Квазігрупа порядку 4 задана за допомогою лише десяти бітів, а для її латинського квадрата потрібно 32 біти.)

Алгоритми шифрування у конструкції BCDMPQ та розшифрування використовують 16 квазігруп: Q_1, Q_2. Q_8. Це у різних кроках. Ці матриці визначаються за допомогою ключа повного і ключа, який генерується з секретним ключем K і складається з 128 біт.

Ключ довжиною 128 біт розподіляється так:

- 16 біт для лідерів 11, 12, ..., 18 (по два біти на кожного лідера)
- 56 біт для квазігруп Q_1, Q_2..... Q_8 (7 біт на кожну квазігрупу, фактично значення порядкового номера)
- 56 біт для квазігруп T_1, T_2,..., T_8 (7 біт на кожну квазігрупи)

Конструкція цього блокового шифру базується на трьох алгоритмах: генерація круглого ключа, шифрування та розшифрування.

Позначимо через K секретний симетричний ключ довжиною 128 біт. Щоб згенерувати повний ключ із секретного ключа, спочатку визначається фіксована довільна квазігрупа Q і фіксований лідер, наприклад l = 0 = [0,0]. Повний ключ отримано шляхом електронних перетворень. Процедура генерації повного ключа описана в алгоритмі Round Key Generation.

Довжина блоку повідомлення BCDMPQ може бути 8n для будь-якого n, для будь якого n, де n = 8, розглядається полегшена версія шифру. Отже, текстове повідомлення розбито на блоки по 64 біта. Після цього алгоритм шифрування застосовується до кожного блоку. Зауважимо що, якщо довжина повідомлення не ділиться на 64, то буде застосовано відповідне доповнення. Алгоритм шифрування складається з двох кроків: Перший ─ використовує матриці Q_1, Q_2, Q_8, а в Другий крок ─ матриці T_1, T_2, ..., T_8.

Інакше, на першому кроці розділяється 64-бітний блок на 8 менших блоків (міні-блоки) з 8 біт. Застосовується електронна трансформація до кожного з цих міні-блоків за допомогою а інший лідер і інша квазігрупа. Фактично, використано лідер l; і квазігрупи Q для і-го міні-блоку. Отриманий рядок використовується як вхідні дані в наступний крок.

На другому кроці застосовано електронні перетворення до кожного отриманого рядка, повторюючи їх 8 разів з поперемінною зміною напрямку. У і-му перетворенні ми використовуєм:

Алгоритм RoundKeyGeneration

Вхід: Секретний ключ $K = K_1K_2 ...K_{128}$, K_i є бітами.

Вихід: Повниий ключ $key = k_1k_2 ...k_{128}$, k_i є бітами.

Ініціалізація: $(Q, *)$ є фіксованою матрицею квазігрупи порядку 4 так що $a * a \neq$ для кожного $a \in Q$, $l = (0, 0)$ є двобітовим.

Для i = 1 до 128

$k_i \leftarrow K_i$;

Для i = 1 до 4

ltmp ← l;

Для j = 1 до 127 крок 2;

tmp ← (k_j , k_{j+1});

$(k_j , k_{j+1}) = m^T + Al^T tmp + Btmp^T + CAl^T tmp \circ CBtmpT$;

$ltmp \leftarrow (k_j , k_{j+1})$;

$ltmp \leftarrow l$;

Для j = 128 до 2 крок 2 ;

$tmp \leftarrow (k_{j-1}, k_j)$;

$(k_{j-1}, k_j) = m^T + Al^T\ tmp + Btmp^T + CAl^T\ tmp \circ CBtmp^T$;

$ltmp \leftarrow (k_{j-1}, k_j)$

квазігрупа T_i , а лідер l_i . Детальний і формалізований алгоритм представлений в Алгоритмі шифрування.

Для цілей розшифрування використано парастроф (Q. \) квазігрупи (Q.*). Отже, потрібно розшифрувати, це означає із зашифрованого тексту повернути електронне перетворення, використовуючи спочатку послідовно квазігрупи T_8, T_7...., T, а потім звернути електронні перетворення міні-блоків (з алгоритму шифрування) з використанням квазігруп Q_8. Q_7..... Q_1. Це можна зробити за допомогою оберненої операції, яку ми згадували раніше. Розшифрування зашифрованого тексту C_1 ,C_2...C_{64} виконується за допомогою алгоритму розшифрування.

Для шифру BCDMPQ були проведені лише попередні дослідження безпеки. Було розглянуто ефект лавини та поширення однобітових і двобітових змін і отримано задовільні результати. Це відкрита дослідницька проблема перевірки стійкості до інших атак блокового шифру.

1.3.6 Потокові шифри

Потокові шифри шифрують дані по одному біту або символу за раз. Існують два типи потокових шифрів: синхронні та асинхронні. Синхронний шифр генерує потік ключів незалежно від вхідних даних, тоді як асинхронний шифр залежить від попередніх зашифрованих символів. Синхронний потоковий шифр часто використовує XOR для змішування вихідних даних з ключовим потоком.

Одним з прикладів потокового шифру є Edon80 – двійковий адитивний шифр, який став фіналістом конкурсу eSTREAM. Цей шифр може працювати в трьох режимах: налаштування ключа, налаштування вектора ініціалізації та генерування потоку ключів:

Алгоритм шифрування

Вхід: круглий ключ = $k_1k_2...k_{128}$, k; є бітами, повідомлення відкритого тексту a = a1a2...a64 є бітами.

Вихід: зашифроване повідомлення C = $C_1C_2...C_{64}$.

Ініціалізація: поставте $l_i = (k_{2i-1}. k_{2i})$ для i = 1, 2, ..., 8.

Пошук квазігрупи Qi використовуючи двійковий порядковий номер, представлений як ($k_{7i-6}. k_{7i-5}$….. $k7_1$), де i = 1, 2, ..., 8. Ініціалізуйте матриці AQ і BQ, а також вектор mQ, для i = 1 , 2, ….8.

Знайдіть квазігрупу 7, використовуючи двійковий порядковий номер, представлений як ($k_{7(i+8)-6}$, $k_{7(i+8)-5}$...., $k_{7(+8))}$. Ініціалізуйте матриці ATi і BT, а також вектор mT для i = 1, 2, ..., 8.

Для i = 1 до 8

ltmp ← l_i;

Для j = 1 до 7 крок 2

tmp ← (a_j , a_{j+1});

(c_j , c_{j+1}) = mT Q_i + AQ_i l T tmp + BQI tmpT +CAQ_i l T tmp ◦ CBQi tmpT ;

ltmp ← (c_j , c_{j+1});

Для i = 1 до 4

ltmp ← l_i;

Для j = 1 до 63 крок 2

tmp ← (c_j , c_{j+1});

(c_j , c_{j+1}) = m T/T_i + AT_i l T tmp + BT_i tmpT + CAT_i l T tmp ◦ CBT_i tmpT;

ltmp ← (c_j , c_{j+1});

ltmp ← l_{i+4};

Для j = 64 до 2 крок 2

tmp ← (c_{j-1}, c_j);

(c_{j-1}, c_j) = mT Ti+4 + ATi+4 l T tmp + BTi+4 tmpT + CATi+4 l T

tmp ◦ CBT_i+4 tmp_T ;

ltmp ← (c_{j-1}, c_j);

Існує кілька дизайнів потокового шифру, заснованого на квазігрупах, і тут ми розглянемо один з них.

Обчислення загальної кількості квазігруп (або ж латинських квадратів) порядку пескладною комбінаторною задачею. На сьогодні До 11-го порядку вдалося визначити кількість скінчених квазігруп з точністю до ізоморфізму (тобто алгебраїчні структури розглядаються як еквівалентні або тотожні в тому сенсі, що їхні алгебраїчні властивості не враховуються, якщо існує ізо морфізм між ними):

$$q^{11} = 19464657391668924966791023043937578299025 \approx e^{e^{4.5}} > 19 \cdot 10^{39}$$

Настільки широкий спектр квазігруп відкриває нові можливості для роз робки криптостійких, ефективних та вдосконалених алгоритмів що мають високий рівень захисту від потенційних атак.

Станом на зараз, існує, як не дивно, досить невелика кількість крипто-графічних алгоритмів, які використовують принципи та властивості квазі груп. В роботах [14] [15] [16] наведені та описані різні способи практи чного застосування квазігруп у криптографії. В [14] Марковський С. пропонує алгоритм квазігрупової обробки рядків та вводить поняття квазігрупо ве перетворення множини Q^+ всіх кінцевих наборів букв кінцевого алфавіту Q(|Q| ≥ 2) : Q+ = x_1 ...x_k|x_i ∈ Q,k ≥ 1, вона є множиною усіх скінченних рядків на

множині Q. Припускаючи, що (Q,·) ─ задана квазігрупа, для сталої літери a ∈ Q визначаються такі перетворення:

$$E = E_{(1)}{}^{a} : Q^{+} \to Q^{+}; D = D^{(1)}{}_{a} : Q^{+} \to Q^{+}$$

Через

$$E_a^{(1)}(x_1 \dots x_k) = y_1 \dots y_k \begin{cases} y_1 = a \cdot x_1, \\ y_{i+1} = y_1 \cdot x_{i+1}, (i = 1, \dots, k-1), \end{cases}$$

$$D_a^{(1)}(y_1 \dots y_k) = x_1 \dots x_k \begin{cases} x_1 = a \setminus y_1, \\ x_{i+1} = y_1 \setminus y_{i+1}, (i = 1, \dots, k-1), \end{cases}$$

де $x_i, y_i \in Q, k \geq 1$

Із властивостей робимо висновок, що перетворення Е та Д є бієкціями і для кожної літери а є дійсним

$$D(E(a)) = a = E(D(a)),$$

іншими словами, $D=E^{-1}$ є зворотною бієкцією Е.

При заданих квазігрпових операціях $*_1$, $*_2$.....$*_n$ на множині Q можна визначити відображення

$$E_1, E_2, \dots, E_n, D, D2, \dots, D_n$$

в такий ж самий через визначення сталих елементів $a_1, a_2, \dots, a_n, \in Q$ (такі що E_i та D. відповідають ($*_n$, a_n). Як наслідок, маємо:

$$E = E(a_n, \dots, a_1)(N) = E_n \circ E_{n-1} \circ \dots \circ E_1, D = D^{(N)}{}_{an,\dots,a1} = D_1 \circ D_2 \circ \dots \circ D_n,$$

де ◦ є звичайною композицією відображень. Квазігрупові операції допускаються бути однаковими. Перетворення $E(a_n,\dots,a_1)^{(N)}$використовується

для шифрування інформації, а в якості ключів автор пропонує використовувати операції $*_1$.

У роботі робиться висновок, що перетворення мають ва жливі криптографічні якості, а саме що кількість квазігрупових операцій на множині C_e величезною (наприклад, існує більше ніж 10^{58000} для |Q|= 256), що забезпечує стійкість до атак грубої сили та статистичних атак. Алгоритм може знайти практичне застосування в проектуванні безпечних баз даних, псевдовипадкових генераторах та ін.[17].

У роботі пропонується блоковий алгоритм для шифрування/дешифрування повідомлень. Це сімейство систем шифрування, параметризованих оптимальною квазігрупою. Сімейство складається з $C_1^{16}!$ систем шифрування. оскільки існує $C_1^{16}!$ способів вибору оптимальної квазігрупи з 16! оптимальних квазігруп. Оптимальною квазігрупою в роботі називається такий групоїд $Q = (Z_{2m}, *)$, в якому виконуються наступні умови:

а) Q є квазігрупою.

б) Кожен рядок та стовпець Q є оптимальним S-боксом розрядності m×m бітів.

Після встановлення системи шифрування здійснюється встановлення на бору із 16 оптимальних квазігруп порядку 16, які вона (система шифрування) використовує. Ці 16 оптимальних квазігруп генеруються на основі використаної оригінальної оптимальної квазігрупи. Крім того, ці оптимальні квазігрупи побудовані на основі 16 оптимальних S-боксів 4×4 біти. Для шифрування де використано повідомлення довжиною 128 біт запропонований шифр виконує загалом 17 раундів, і кожен раунд використовує 128-бітовий раундовий ключ разом з одним нелінійним перетворенням і двома лінійними перетвореннями. Нелінійне перетворення є не що інше, як квазігрупова операція, заснована на залежному від ключа S-боксу [18]. Тобто для кожної квазігрупової операції нелінійне перетворення залежить від круглого ключа та вибирає один S-боксу із 16 S-боксів.

Шістнадцять оптимальних квазігруп порядку 16 використовуються в

проектуванні запропонованого блокового шифру. Таким чином, усі операції виконуються у формі 4-бітових (також званих ніблів (англ, nibbles) агрегацій. Нехай кожен байт даних буде розділений на два нібли. Тобто байтове значення представляється як $x = x_1x_0$, де x_1 і x_0 є ніблами [19]. Тоді квазігрупові операції для шифрування та розшифрування визначаються як

$$x_1x_0 *i\ y_1y_0 = (x_1 \cdot_i y_i)||(x_0 \cdot_i y_0)$$

де

$$x_1x_0 \circ i\ y_1y_0 = (x_1 \backslash_i y_i)||(x_0 \backslash_i y_0)$$

відповідно, де $(\cdot_i, \backslash_i)$ та $(*_i, \circ_i)$ є квазігруповими операціями відповідно до ніблів, $0 \leq 1 < 15$ і $||$ є кроками, які сумують два 4-бітних значення в 8-бітне. Також, квазігрупа, називається оберненою (або оберненою зліва) квазігруповою операцією відносно символу операції. Тобто, якщо квазігрупа $Q_i = (Z_n, *i)$ використовується для шифрування, то її оборотна зліва квазігрупа $LIQ_i = (Z_n, \circ_i)$ використовується для розшифрування.

В роботі також було проаналізовано стійкість шифру до ряду атак, включаючи лінійний та диференційний криптоаналіз, в результаті чого було доведено його стійкість [20].

2 МЕТОД ШИФРУВАННЯ НА ОСНОВІ КВАЗІГРУПОВОЇ ТОТОЖНОСТІ

2.1 Метод зашифрування за І законом Шредера

І закон Шредера:

$$(x \cdot y) \cdot y = x \cdot (x \cdot y)$$

означає, що для будь-яких елементів x і y квазігрупи операції над ними задовольняють таку властивість: результат множення елементів x і y, а потім знову множення результату на y, дорівнює результату множення x на результат операції $x \cdot y$. В цьому законі важливо, що одна й та сама операція виконується з елементами в різному порядку, але результат залишається однаковим.

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

Рисунок 2.1 ─ Латинський квадрат 8-го порядку

Метод шифрування за допомогою І закону Шредера:

Вхідні дані:

- Множина символів повідомлення M ={ $m_1\, m_2\, m_3\, m_4\, \ldots\, m_p$ }
- Множина ключів K={ $k_1\, k_2\, k_3\, k_4\, \ldots\, k_s$ }

Оскільки використовується латинський квадрат 8-го порядку, то символи повідомлення m_i (i=1…p) та символи секретного ключа k_j (j=1…s) приймають значення від 0 до 7. Це означає, що двійковий код повідомлення M та секретного ключа K розбивається на групи по 3 розряди, кожна з яких і

відповідає символам зі значеннями від 0 до 7.

Шифрування кожного символу повідомлення здійснюється таким чином:

1. Маємо інформаційний рядок, призначений для шифрування:

$m_1\, m_2\, m_3\, m_4\, \ldots\, m_p$

2. Вибираємо елемент ключа k_j з множини K , і виконуємо квазігрупову операцію, яка задовольняє I закон Шредера. Для кожного символу повідомлення m_i, застосовується операція шифрування:

$$(k_j \cdot m_i)\, m_i = k_j \cdot (k_j \cdot m_i)$$

Тобто шифрування кожного символу відбувається за допомогою двох кроків, де на кожному кроці виконується та сама операція, але між різними значеннями: спершу множимо ключ k_j на повідомлення m_i , а потім отриманий результат множимо на m_i ще раз. Це перетворює повідомлення, шифруючи його через ключ і операції з ним. А саме, маючи набір ключів та інформаційний рядок:

$$k_1\, k_2\, k_3\, k_4\, \ldots\, k_s \,||\, m_1\, m_2\, m_3\, m_4\, \ldots\, m_p$$

шифрування здійснюється поелементно, тобто:

$$\leftarrow k_1\, k_2\, k_3\, k_4\, \ldots\, k_s \,||\, m_1\, m_2\, m_3\, m_4\, \ldots\, m_p$$

згідно з законом I Шредера спочатку виконується операція $L(k_1,m_1) \rightarrow c_1^*$, а потім ця ж сама операція, але між іншими елементами, а саме $L(c_1^*,m_1) \rightarrow c_1$. Таким чином шифрується перший символ.

В результаті, наступним кроком шифрування буде здійснюватися, за таким набором:

$$k_2\, k_3\, k_4\, \ldots\, k_s\, m_1 \,||\, m_2\, m_3\, m_4\, \ldots\, m_p$$

оскільки шифрування здійснюється поелементно, тобто зсув для наступних символів:

$$\leftarrow k_2\, k_3\, k_4\ \ldots\ k_s\ \ m_1 || \, m_2\, m_3\, m_4\ \ldots\ m_p$$

Тут теж, згідно з законом І Шредера, спочатку виконується операція $L(k_2,m_2) \rightarrow c_2^*$, а потім ця ж сама операція, але між іншими елементами, а саме $L(c_2^*,m_2) \rightarrow c_2$. Таким чином шифрується другий символ повідомлення.

В результаті, наступним кроком шифрування буде здійснюватися, за таким набором:

$$k_3\, k_4\ \ldots\ k_s\ \ m_1\, m_2\, ||m_3\, m_4\ \ldots\ m_p$$

оскільки шифрування здійснюється поелементно, тобто зсув для наступних символів:

$$\leftarrow k_3\, k_4\ \ldots\ k_s\ \ m_1\, m_2\, ||m_3\, m_4\ \ldots\ m_p$$

Тут теж, згідно з законом І Шредера, спочатку виконується операція $L(k_3,m_3) \rightarrow c_3^*$, а потім ця ж сама операція, але між іншими елементами, а саме $L(c_3^*,m_3) \rightarrow c_3$. Таким чином шифрується третій символ повідомлення.

В результаті, наступним кроком шифрування буде здійснюватися, за таким набором:

$$k_4\ \ldots\ k_s\ \ m_1\, m_2\, m_3\, ||m_4\ \ldots\ m_p$$

оскільки шифрування здійснюється поелементно, тобто зсув для наступних символів:

$$\leftarrow k_4 \ \dots \ k_s \, m_1 \, m_2 \, m_3 \, || m_4 \ \dots \ m_p$$

Тут теж, згідно з законом І Шредера, спочатку виконується операція $L(k_4,m_4) \rightarrow c_4^*$, а потім ця ж сама операція, але між іншими елементами, а саме $L(c_4^*,m_4) \rightarrow c_4$. Таким чином шифрується четвертий символ повідомлення.

Таким самим чином шифруються і так далі всі інші символи повідомлення, тобто передостаннім кроком шифрування буде здійснюватися, за таким набором:

$$k_s \ \ m_1 \, m_2 \, m_3 m_4 \ \ || \dots \ m_p$$

оскільки шифрування здійснюється поелементно, тобто зсув для наступних символів:

$$\leftarrow k_s \ \ m_1 \, m_2 \, m_3 \, m_4 \ || \dots \ m_p$$

Тут теж, згідно з І законом Шредера, спочатку виконується операція $L(k_s,m_p) \rightarrow c_p^*$, а потім ця ж сама операція, але між іншими елементами, а саме $L(c_p^*,m_p) \rightarrow c_p$. Таким чином шифрується передостанній символ повідомлення, який використовує ключ. Всі наступні елементи шифрування будуть здійснюватися між символами самих повідомлень, а саме, наступний крок шифрування буде здійснюватися за таким набором:

$$m_1 \, m_2 \, m_3 \, m_4 \ \dots \ m_p || m_{p+1}$$

оскільки шифрування здійснюється поелементно, тобто є зсув для наступних символів:

$$\leftarrow m_1 \, m_2 \, m_3 \, m_4 \ \dots \ m_p || m_{p+1}$$

Тут теж, згідно з І законом Шредера, спочатку виконується операція $L(m_1,m_{p+1}) \rightarrow c^*_{p+1}$, а потім ця ж сама операція, але між іншими елементами, а саме $L(c^*_{p+1}, m_{p+1}) \rightarrow c_{p+1}$. Аналогічно шифрується останній символ повідомлення.

Таким чином шифрована послідовність згідно І закону Шредера має загальний вигляд:

$$(k_1 \cdot m_1)\, m_1, (k_2 \cdot m_2)\, m_2, (k_3 \cdot m_3)\, m_3, \ldots, (k_s \cdot m_p)\, m_p, \ldots, (m_1 \cdot m_{p+1})\, m_{p+1} \ldots$$

Отже, множина зашифрованого повідомлення $C=\{c_1, c_2, c_3, c_3, \ldots, c_{p+1}\}$.

2.1.1 Загальна схема зашифрування за квазігруповою тотожністю І-го закону Шредера

Вхідні дані:

- Множина символів повідомлення $M = \{ m_1\, m_2\, m_3\, m_4 \ldots m_p \}$
- Множина ключів $K = \{ k_1\, k_2\, k_3\, k_4 \ldots k_s \}$

Проміжні обчислення:

- L це таблиця Келі квазігрупи за І законом Шредера
- Множина проміжних шифрованих значень $C=\{c^*_1, c^*_2, c^*_3, c^*_4, \ldots, c^*_{p+1}\}$

Вихідні дані:

- Множина зашифрованого повідомлення $C=\{c_1, c_2, c_3, c_3, \ldots, c_{p+1}\}$

$k_1\, k_2\, k_3\, k_4 \ldots k_s \,||\, m_1\, m_2\, m_3\, m_4 \ldots m_p$

$L(k_1,m_1) \rightarrow c^*_1, L(c^*_1,m_1) \rightarrow c_1$

$\leftarrow k_1\, k_2\, k_3\, k_4 \ldots k_s \,||\, m_1\, m_2\, m_3\, m_4 \ldots m_p$

$k_2\, k_3\, k_4 \ldots k_s\, m_1 || m_2\, m_3\, m_4 \ldots m_p$

$L(k_2,m_2) \rightarrow c^*_2, L(c^*_2,m_2) \rightarrow c_2$

$\leftarrow k_2\, k_3\, k_4 \ldots k_s\ m_1 || m_2\, m_3\, m_4 \ldots m_p$

$k_3 k_4 \dots k_s \; m_1 m_2 || m_3 m_4 \dots m_p$

$L(k_3,m_3) \rightarrow c_3^*, L(c_3^*,m_3) \rightarrow c_3$

$\leftarrow k_3 k_4 \dots k_s \; m_1 m_2 || m_3 m_4 \dots m_p$

$k_4 \dots k_s \; m_1 m_2 m_3 || m_4 \dots m_p$

$L(k_4,m_4) \rightarrow c_4^*, L(c_4^*,m_4) \rightarrow c_4$

$\leftarrow k_4 \dots k_s \; m_1 m_2 m_3 || m_4 \dots m_p$

...

$k_s \; m_1 m_2 m_3 m_4 || \dots m_p$

$L(k_s,m_p) \rightarrow c_p^*, L(c_p^*,m_p) \rightarrow c_p$

$\leftarrow k_s \; m_1 m_2 m_3 m_4 || \dots m_p$

$m_1 m_2 m_3 m_4 \dots m_p || m_{p+1}$

$L(m_1,m_{p+1}) \rightarrow c_{p+1}^*, L(c_{p+1}^*,m_{p+1}) \rightarrow c_{p+1}$

...

Приклад:

У латинаському квадраті стопцями є елементи ключа k_j , а рядками є символи повідомлення m_p.

M={015705321432}

01234567||015705321432

0·0=1,

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

1·0=0

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$C_1=0$

12345670||15705321432

1·1=2,

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

2·1=1

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$C_2=1$

23456701||5705321432

2·5=7,

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$7 \cdot 5 = 0$

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$C_3 = 0$

34567015||705321432

$3 \cdot 7 = 6$,

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$6 \cdot 7 = 1$

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$C_4=1$

45670157||05321432

$4\cdot 0=4$,

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$4\cdot 0=4$

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$C_5=4$

56701570||5321432

5·5=2,

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

2·5=7

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

C_6=7

67015705||321432

6·3=5,

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

5·3=4

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$C_7=4$

70157053||21432

$7\cdot 2=5$,

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$5\cdot 2=7$

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$C_8=7$

01570532||1432

$0\cdot 1=0$,

*	0	1	2	3	4	5	6	7
0	0	3	2	4	5	6	7	
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$0 \cdot 1 = 0$

*	0	1	2	3	4	5	6	7
0	0	3	2	4	5	6	7	
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$C_9 = 0$

15705321||432

$1 \cdot 4 = 5,$

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$5 \cdot 4 = 1$

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$C_{10}=1$

57053214||32

$5 \cdot 3=4$,

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$4 \cdot 3=7$

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$C_{11}=7$

70532143||2

7·2=5,

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

5·2=7

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

$C_{12}=7$

Отже, C={010147470177}

2.2 Метод розшифрування за І законом Шредера

Метод розшифрування:

Для розшифрування використовуємо ту саму множину ключів k . Оскільки І закон Шредера стверджує, що обидві сторони рівняння рівні, то можна повернути початковий символ повідомлення за допомогою оборотних операцій. Оскільки квазігрупа яка задовольняє перший закон Шредера є тотально-симетричною, то це означає, що таблиця Келі, для кожної оборотної квазігрупової операції є однаковою. Це дає можливість застосовувати цю ж саму таблицю Келі, що використовували для зашифрування, використати її і для розшифрування.

Вхідні дані:

- Множина символів зашифрованого повідомлення C={c_1 ,c_2 ,c_3 ,c_3,…,c_{p+1}}
- Множина ключів K={ k_1 k_2 k_3 k_4 … k_s }

На кожному кроці, застосовуючи квазігрупову операцію з тим же набором ключів, повертаємося до початкового значення символа m_i. А саме, розшифрування кожного символу відбувається за допомогою двох кроків, де на першому кроці виконується та сама операція між ключем k_j та символом зашифрованого повідомлення c_i , а на другому кроці отриманий результат m_i^* множимо на c_i ще раз. Це перетворює повідомлення на вхідний інформаційний рядок з m_i . А саме, розшифрування здійснюється так набір ключів та інформаційний рядок є вхідними даними, тобто:

$$k_1 k_2 k_3 k_4 \ \dots \ k_s \,||\, c_1 c_2 c_3 c_4 \ \dots \ c_p$$

розшифрування здійснюється поелементно, тобто:

$$\leftarrow k_1 k_2 k_3 k_4 \ \dots \ k_s \,||\, c_1 c_2 c_3 c_4 \ \dots \ c_p$$

згідно з I законом Шредера спочатку виконується операція $L(k_1,c_1) \rightarrow m_1^*$,а потім ця ж сама операція, але між іншими елементами, а саме $L(m_1^*,c_1) \rightarrow m_1$. Таким чином розшифровується перший символ.

В результаті, наступним кроком розшифрування буде здійснюватися, за таким набором:

$$k_2 k_3 k_4 \ \dots \ k_s \, c_1 \,||\, c_2 c_3 c_4 \ \dots \ c_p$$

оскільки розшифрування здійснюється поелементно, тобто:

$$\leftarrow k_2 k_3 k_4 \ \dots \ k_s \, c_1 || c_2 c_3 c_4 \ \dots \ c_p$$

згідно з I законом Шредера спочатку виконується операція $L(k_2,c_2) \rightarrow m_2^*$,а

потім ця ж сама операція, але між іншими елементами, а саме $L(m_2^*,c_2) \rightarrow m_2$. Таким чином розшифровується другий символ.

В результаті, наступним кроком розшифрування буде здійснюватися, за таким набором:

$$k_3\, k_4\, \ldots\, k_s\, c_1 c_2 \,||\, c_3\, c_4\, \ldots\, c_p$$

оскільки розшифрування здійснюється поелементно, тобто:

$$\leftarrow k_3\, k_4\, \ldots\, k_s\, c_1 c_2\, ||c_3\, c_4\, \ldots\, c_p$$

згідно з І законом Шредера спочатку виконується операція $L(k_3,c_3) \rightarrow m_3^*$,а потім ця ж сама операція, але між іншими елементами, а саме $L(m_3^*,c_3) \rightarrow m_3$. Таким чином розшифровується третій символ.

В результаті, наступним кроком розшифрування буде здійснюватися, за таким набором:

$$k_4\, \ldots\, k_s\, c_1 c_2\, c_3\, ||\, c_4\, \ldots\, c_p$$

оскільки розшифрування здійснюється поелементно, тобто:

$$\leftarrow k_4\, \ldots\, k_s\, c_1 c_2\, c_3 ||\, c_4\, \ldots\, c_p$$

згідно з І законом Шредера спочатку виконується операція $L(k_4,c_4) \rightarrow m_4^*$,а потім ця ж сама операція, але між іншими елементами, а саме $L(m_4^*,c_4) \rightarrow m_4$. Таким чином розшифровується четвертий символ.

В результаті, наступним кроком розшифрування буде здійснюватися, за таким набором:

$$k_s\, c_1 c_2\, c_3\, c_4\, ||\, \ldots\, c_p$$

оскільки розшифрування здійснюється поелементно, тобто:

$$\leftarrow k_s\, c_1 c_2\, c_3\, c_4 \,||\, \dots\, c_p$$

згідно з І законом Шредера спочатку виконується операція $L(k_s,c_p) \rightarrow m_p^*$,а потім ця ж сама операція, але між іншими елементами, а саме $L(m_p^*,c_p) \rightarrow m_p$. Таким чином розшифровується передостанній символ повідомлення, який використовує ключ. Всі наступні елементи шифрування будуть здійснюватися між символами самих повідомлень, а саме, наступний крок шифрування буде здійснюватися за таким набором:

$$c_1 c_2\ c_3\, c_4 \dots\, c_p \,||\, c_{p+1}$$

оскільки розшифрування здійснюється поелементно, тобто:

$$\leftarrow c_1 c_2\, c_3\, c_4 \dots\, c_p \,||\, c_{p+1}$$

згідно з І законом Шредера спочатку виконується операція $L(c_1,c_{p+1}) \rightarrow m_{p+1}^*$, а потім ця ж сама операція, але між іншими елементами, а саме $L(m_{p+1}^*,c_{p+1}) \rightarrow m_{p+1}$. Таким чином розшифровується останній символ.

Таким чином розшифрована послідовність згідно І закону Шредера має загальний вигляд:

$$(k_1 \cdot c_1)\, c_1,(k_2 \cdot c_2)\, c_2,(k_3 \cdot c_3)\, c_3,\dots,(k_s \cdot c_p)\, c_p,\dots,(c_1 \cdot c_{p+1})\, c_{p+1}\dots$$

Отже, множина розшифрованого повідомлення M =$\{ m_1\, m_2\, m_3\, m_4 \dots m_p \}$

2.2.1 Загальна схема розшифрування за квазігруповою тотожністю І-го закону Шредера

Вхідні дані:

- Множина символів зашифрованого повідомлення C={c_1 ,c_2 ,c_3 ,c_3,…,c_{p+1}}
- Множина ключів K={ k_1 k_2 k_3 k_4 … k_s }

Проміжні обчислення:

- L це таблиця Келі квазігрупи за І законом Шредера
- Множина проміжних шифрованих значень M={m_1^* , m , m_3^*, m_4^*,…, m_{p+1}^*}

Вихідні дані:

- Множина розшифрованого повідомлення M ={ m_1 m_2 m_3 m_4 … m_p }

$k_1\,k_2\,k_3\,k_4\ \ldots\ k_s \,||\, c_1\,c_2\,c_3\,c_4\ \ldots\ c_p$

$L(k_1,c_1) \rightarrow m_1^*$, $L(m_1^*,c_1) \rightarrow m_1$

$\leftarrow k_1\,k_2\,k_3\,k_4\ \ldots\ k_s \,||\, c_1\,c_2\,c_3\,c_4\ \ldots\ c_p$

$k_2\,k_3\,k_4\ \ldots\ k_s\ \ c_1 || \,c_2\,c_3\,c_4\ \ldots\ c_p$

$L(k_2,c_2) \rightarrow m_2^*$, $L(m_2^*,c_2) \rightarrow m_2$

$\leftarrow k_2\,k_3\,k_4\ \ldots\ k_s\ \ c_1 || \,c_2\,c_3\,c_4\ \ldots\ c_p$

$k_3\,k_4\ \ldots\ k_s\ \ c_1\,c_2 \,||\, c_3\,c_4\ \ldots\ c_p$

$L(k_3,c_3) \rightarrow m_3^*$, $L(m_3^*,c_3) \rightarrow m_3$

$\leftarrow k_3\,k_4\ \ldots\ k_s\ \ c_1 c_2 \,||\, c_3\,c_4\ \ldots\ c_p$

$k_4\ \ldots\ k_s\ \ c_1\,c_2 c_3 \,||\, c_4\ \ldots\ c_p$

$L(k_4,c_4) \rightarrow m_4^*$, $L(m_4^*,c_4) \rightarrow m_4$

$\leftarrow k_4\ \ldots\ k_s\,c_1\,c_2\,c_3 \,||\, c_4\ \ldots\ c_p$

...

$k_s\ \ c_1\,c_2\,c_3\,c_4 || \ \ldots\ c_p$

$L(k_s,c_p) \rightarrow m_p^*$, $L(m_p^*,c_p) \rightarrow m_p$

$\leftarrow k_s\,c_1 c_2\,c_3 c_4 \ ||\ \ldots\ c_p$

$c_1\,c_2\,c_3\,c_4\ \ldots\ c_p || \ c_{p+1}$

$L(c_1,c_{p+1}) \rightarrow m_{p+1}^*$, $L(m_{p+1}^*,c_{p+1}) \rightarrow m_{p+1}$

...

Приклад:
C={010147470177}
01234567||010147470177
0·0=1, 1·0=0
C_1=0
12345670||10147470177
1·1=2, 2·1=1
C_2=1
23456701||0147470177
2·0=5, 5·0=5
C_3=5
34567010||147470177
3·1=6, 6·1=7
C_4=7
45670101||47470177
4·4=5, 5·4=0
C_5=0
56701014||7470177
5·7=2, 2·7=5
C_6=5
67010147||470177
6·4=7, 7·4=3
C_7=3
70101474||70177
7·7=2, 2·7=2
C_8=2
01014747||0177
0·0=2, 2·0=1
C_9=1
10147470||177
1·1=2, 2·1=4
C_{10}=4
01474701||77
0·7=7, 7·7=3
C_{11}=3
14747017||7
1·7=4, 4·7=2
C_{12}=2
Отже, M={015705321432}

2.3 Метод зашифрування за II законом Шредера

В теорії алгебричних структур II закон Шредера визначається квазігруповою тотожністю:

$$xy \cdot yx = y$$

Цей закон означає, що для будь-яких елементів *x* та *y* квазігрупи, коли один елемент множиться на інший за певною операцією, результатом буде сам елемент *y*.

Квазігрупи — це алгебраїчні структури з бінарною операцією (позначена тут як "·"), яка має певні властивості, включаючи право і ліво-оборотність. Для будь-яких елементів *x* і *y* враховуючи виконання операцій за вказаним законом Шредера , виконуючи операцію $x \cdot y$, результатом завжди буде *y*, незалежно від значення *x*. Аналогічно і для операції між елементами *y x*.

*	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	3	0	2	5	6	7	4
2	2	0	3	1	6	7	4	5
3	3	2	1	0	7	4	5	6
4	4	5	6	7	1	0	3	2
5	5	6	7	4	0	3	2	1
6	6	7	4	5	3	2	1	0
7	7	4	5	6	2	1	0	3

Рисунок 2.2 – Нормалізований латинський квадрат 8-го порядку

Нормалізований латинський квадрат це квадрат у якому перший рядок і перший стовпець записані у лексикографігному порядку.

Метод зашифрування:

Вхідні дані:

- Множина символів повідомлення M ={ $m_1\, m_2\, m_3\, m_4\, \ldots\, m_p$ }
- Множина ключів K={ $k_1\, k_2\, k_3\, k_4\, \ldots\, k_s$ }

Оскільки використовується латинський квадрат 8-го порядку, то символи повідомлення m_i (i=1…p) та символи секретного ключа k_j (j=1…s) приймають значення від 0 до 7. Це означає, що двійковий код повідомлення M та секретного ключа K розбивається на групи по 3 розряди, кожна з яких і відповідає символам зі значеннями від 0 до 7.

Шифрування кожного символу повідомлення здійснюється таким чином:

1. Маємо інформаційний рядок, призначений для шифрування:

$$m_1\, m_2\, m_3\, m_4\, \ldots\, m_p$$

2. Вибираємо елемент ключа k_j з множини K , і виконуємо квазігрупову операцію, яка задовольняє II закон Шредера. Для кожного символу повідомлення m_i, застосовується операція шифрування:

$$k_j m_i \cdot m_i k_j = m_i$$

Тобто шифрування кожного символу відбувається за допомогою трьох кроків, де на кожному кроці виконується та сама операція, але між різними значеннями ключа та повідомлення. Проміжними операціями є множення ключа k_j на повідомлення m_i, а потім комутативне множення ключа k_j на повідомлення m_i і отримані результати множимо між собою. Це перетворює повідомлення, шифруючи його через ключ і 3 проміжні операції, які виконуються за однією і тією ж самою функцією. А саме, маючи набір ключів та інформаційний рядок:

$$k_1\, k_2\, k_3\, k_4\, \ldots\, k_s \,||\, m_1\, m_2\, m_3\, m_4\, \ldots\, m_p$$

шифрування здійснюється поелементно, тобто:

$$\leftarrow k_1\, k_2\, k_3\, k_4\, \ldots\, k_s \,||\, m_1\, m_2\, m_3\, m_4\, \ldots\, m_p$$

згідно з законом ІІ Шредера спочатку виконується операція $L(k_1,m_1) \rightarrow c_1^*$, а потім ця ж сама операція, але між іншими елементами, а саме $L(c_1^*,L(m_1,k_1)) \rightarrow c_1$. Таким чином шифрується перший символ.

В результаті, наступним кроком шифрування буде здійснюватися, за таким набором:

$$k_2\, k_3\, k_4\, \ldots\, k_s\, m_1 \,||\, m_2\, m_3\, m_4\, \ldots\, m_p$$

оскільки шифрування здійснюється поелементно, тобто зсув для наступних символів:

$$\leftarrow k_2\, k_3\, k_4\, \ldots\, k_s\;\; m_1 || \, m_2\, m_3\, m_4\, \ldots\, m_p$$

Тут теж, згідно з законом ІІ Шредера, спочатку виконується операція $L(k_2,m_2) \rightarrow c_2^*$, а потім ця ж сама операція, але між іншими елементами, а саме $L(c_2^*, L(m_2,k_2)) \rightarrow c_2$. Таким чином шифрується другий символ повідомлення.

В результаті, наступним кроком шифрування буде здійснюватися, за таким набором:

$$k_3\, k_4\, \ldots\, k_s\;\; m_1\, m_2 \,||\, m_3\, m_4\, \ldots\, m_p$$

оскільки шифрування здійснюється поелементно, тобто зсув для наступних символів:

$$\leftarrow k_3\, k_4\, \ldots\, k_s\;\; m_1\, m_2 \,||\, m_3\, m_4\, \ldots\, m_p$$

Тут теж, згідно з законом ІІ Шредера, спочатку виконується операція $L(k_3,m_3) \rightarrow c_3^*$, а потім ця ж сама операція, але між іншими елементами, а саме $L(c_3^*, L(m_3,k_3)) \rightarrow c_3$. Таким чином шифрується третій символ повідомлення.

В результаті, наступним кроком шифрування буде здійснюватися, за таким набором:

$$k_4 \ldots k_s \; m_1 m_2 m_3 || m_4 \ldots m_p$$

оскільки шифрування здійснюється поелементно, тобто зсув для наступних символів:

$$\leftarrow k_4 \ldots k_s \, m_1 m_2 m_3 || m_4 \ldots m_p$$

Тут теж, згідно з законом II Шредера, спочатку виконується операція $L(k_4,m_4) \rightarrow c_4^*$, а потім ця ж сама операція, але між іншими елементами, а саме $L(c_4^*, L(m_4,k_4)) \rightarrow c_4$. Таким чином шифрується четвертий символ повідомлення.

Таким самим чином шифруються і так далі всі інші символи повідомлення, тобто передостаннім кроком шифрування буде здійснюватися, за таким набором:

$$k_s \; m_1 m_2 m_3 m_4 \; || \ldots m_p$$

оскільки шифрування здійснюється поелементно, тобто зсув для наступних символів:

$$\leftarrow k_s \; m_1 m_2 m_3 m_4 \; || \ldots m_p$$

Тут теж, згідно з II законом Шредера, спочатку виконується операція $L(k_s,m_p) \rightarrow c_p^*$, а потім ця ж сама операція, але між іншими елементами, а саме $L(c_p^* , L(m_p,k_p)) \rightarrow c_p$. Таким чином шифрується передостанній символ повідомлення, який використовує ключ. Всі наступні елементи шифрування будуть здійснюватися між символами самих повідомлень, а саме, наступний

крок шифрування буде здійснюватися за таким набором:

$$m_1 m_2 m_3 m_4 \ \dots \ m_p || m_{p+1}$$

оскільки шифрування здійснюється поелементно, тобто є зсув для наступних символів:

$$\leftarrow m_1 m_2 m_3 m_4 \ \dots \ m_p || m_{p+1}$$

Тут теж, згідно з II законом Шредера, спочатку виконується операція $L(m_1, m_{p+1}) \rightarrow c^*_{p+1}$, а потім ця ж сама операція, але між іншими елементами, а саме $L(c^*_{p+1}, L(m_{p+}1, m1)) \rightarrow c_{p+1}$. Аналогічно шифрується останній символ повідомлення.

Таким чином шифрована послідовність згідно II закону Шредера має загальний вигляд:

$$k_1 m_1 \cdot m_1 k_1, k_2 m2 \cdot m_2 k2, k_3 m_3 \cdot m_3 k_3, k_4 m_4 \cdot m_4 k_4, \dots, k m_k \cdot m_k k, \dots, m_1 m_{p+1} \cdot m_{p+1} m_1$$

Отже, множина зашифрованого повідомлення $C=\{c_1, c_2, c_3, c_3, \dots, c_{p+1}\}$.

2.3.1 Загальна схема зашифрування за квазігруповою тотожністю II-го закону Шредера

Вхідні дані:

- Множина символів повідомлення $M = \{ m_1 m_2 m_3 m_4 \ \dots \ m_p \}$
- Множина ключів $K = \{ k_1 k_2 k_3 k_4 \ \dots \ k_s \}$

Проміжні обчислення:

- L це таблиця Келі квазігрупи за II законом Шредера
- Множина проміжних шифрованих значень $C=\{c^*_1, c^*_2, c^*_3, c^*_4, \dots, c^*_{p+1}\}$

Вихідні дані:

- Множина зашифрованого повідомлення C={c_1 ,c_2 ,c_3 ,c_3,…,c_{p+1}}

$k_1 k_2 k_3 k_4 \ \dots \ k_s \,||\, m_1 m_2 m_3 m_4 \ \dots \ m_p$

$L(k_1,m_1) \rightarrow c_1^*$, $L(c_1^*, L(m_1,k_1)) \rightarrow c_1$.

$\leftarrow k_1 k_2 k_3 k_4 \ \dots \ k_s \,||\, m_1 m_2 m_3 m_4 \ \dots \ m_p$

$k_2 k_3 k_4 \ \dots \ k_s m_1 || m_2 m_3 m_4 \ \dots \ m_p$

$L(k_2,m_2) \rightarrow c_2^*$, $L(c_2^*, L(m_2,k_2)) \rightarrow c_2$.

$\leftarrow k_2 k_3 k_4 \ \dots \ k_s \ m_1 || m_2 m_3 m_4 \ \dots \ m_p$

$k_3 k_4 \ \dots \ k_s \ m_1 m_2 \,||m_3 m_4 \ \dots \ m_p$

$L(k_3,m_3) \rightarrow c_3^*$, $L(c_1^*, L(m_3,k_3)) \rightarrow c_3$.

$\leftarrow k_3 k_4 \ \dots \ k_s \, m_1 m_2 \,||\, m_3 m_4 \ \dots \ m_p$

$k_4 \ \dots \ k_s \ m_1 m_2 m_3 \,||m_4 \ \dots \ m_p$

$L(k_4,m_4) \rightarrow c_4^*$,$L(c_4^*, L(m_4,k_4)) \rightarrow c_4$

$\leftarrow k_4 \ \dots \ k_s \ m_1 m_2 m_3 \,||\, m_4 \ \dots \ m_p$

...

$k_s \ m_1 m_2 m_3 m_4 \,||\, \dots \ m_p$

$L(k_s,m_p) \rightarrow c_p^*$, $L(c_p^*, L(m_p, k_s)) \rightarrow c_p$

$\leftarrow k_s \ m_1 m_2 m_3 m_4 \,||\, \dots \ m_p$

$m_1 m_2 m_3 m_4 \ \dots \ m_p || \ m_{p+1}$

$L(m_1,m_{p+1}) \rightarrow c_{p+1}^*$,$L(c_{p+1}^*, L(m_{p+}1,m1)) \rightarrow c_{p+1}$

...

2.4 Метод розшифрування за II законом Шредера

Метод розшифрування:

На кожному кроці, застосовуючи квазігрупову операцію з тим же набором ключів, повертаємося до початкового значення символа m_i. А саме, розшифрування кожного символу відбувається за допомогою трьох кроків, де на кожному кроці виконується та сама операція, але між різними значеннями ключа та повідомлення. Проміжними операціями є множення ключа k_j на

повідомлення c_i, а потім комутативне множення ключа k_j на повідомлення c_i і отримані результати множимо між собою. Це перетворює повідомлення, розшифровуючи його через ключ і 3 проміжні операції, які виконуються за однією і тею ж самою функцією.

Вхідні дані:

- Множина символів зашифрованого повідомлення $C=\{c_1, c_2, c_3, c_3, \ldots, c_{p+1}\}$
- Множина ключів $K=\{ k_1 k_2 k_3 k_4 \ \ldots \ k_s \}$

Маючи набір ключів та інформаційний рядок:

$$k_1 k_2 k_3 k_4 \ \ldots \ k_s \,||\, c_1 c_2 c_3 c_4 \ \ldots \ c_p$$

розшифрування здійснюється поелементно, тобто:

$$\leftarrow k_1 k_2 k_3 k_4 \ \ldots \ k_s \,||\, c_1 c_2 c_3 c_4 \ \ldots \ c_p$$

згідно з II законом Шредера спочатку виконується операція $L(k_1,c_1) \rightarrow m_1^*$, а потім ця ж сама операція, але між іншими елементами, а саме $L(m_1^*, L(c_1,k_1)) \rightarrow m_1$. Таким чином розшифровується перший символ.

В результаті, наступним кроком розшифрування буде здійснюватися, за таким набором:

$$k_2 k_3 k_4 \ \ldots \ k_s \, c_1 \,||\, c_2 c_3 c_4 \ \ldots \ c_p$$

оскільки розшифрування здійснюється поелементно, тобто:

$$\leftarrow k_2 k_3 k_4 \ \ldots \ k_s \, c_1 || c_2 c_3 c_4 \ \ldots \ c_p$$

згідно з II законом Шредера спочатку виконується операція $L(k_2,c_2) \rightarrow m_2^*$, а потім ця ж сама операція, але між іншими елементами, а саме

L(m_2^*,L(c_2,k_2))→m_2. Таким чином розшифровується другий символ.

В результаті, наступним кроком розшифрування буде здійснюватися, за таким набором:

$$k_3\, k_4\, \ldots\, k_s\, c_1 c_2\, ||\, c_3\, c_4\, \ldots\, c_p$$

оскільки розшифрування здійснюється поелементно, тобто:

$$\leftarrow k_3\, k_4\, \ldots\, k_s\, c_1 c_2\, ||c_3\, c_4\, \ldots\, c_p$$

згідно з II законом Шредера спочатку виконується операція L(k_3,c_3) → m_3^* ,а потім ця ж сама операція, але між іншими елементами, а саме L(m_3^*,L(c_3,k_3)))→m_3. Таким чином розшифровується третій символ.

В результаті, наступним кроком розшифрування буде здійснюватися, за таким набором:

$$k_4\, \ldots\, k_s\, c_1 c_2\, c_3\, ||\, c_4\, \ldots\, c_p$$

оскільки розшифрування здійснюється поелементно, тобто:

$$\leftarrow k_4\, \ldots\, k_s\, c_1 c_2\, c_3 ||\, c_4\, \ldots\, c_p$$

згідно з II законом Шредера спочатку виконується операція L(k_4,c_4) → m_4^* ,а потім ця ж сама операція, але між іншими елементами, а саме L(m_4^*,L(c_4,k_4))→m_4. Таким чином розшифровується четвертий символ.

В результаті, наступним кроком розшифрування буде здійснюватися, за таким набором:

$$k_s\, c_1 c_2\, c_3\, c_4\, ||\, \ldots\, c_p$$

оскільки розшифрування здійснюється поелементно, тобто:

$$\leftarrow k_s\, c_1 c_2\, c_3\, c_4 \,||\, \dots\, c_p$$

згідно з II законом Шредера спочатку виконується операція $L(k_s,c_p) \rightarrow m_p^*$,а потім ця ж сама операція, але між іншими елементами, а саме $L(\, m_p^*\, ,L(c_p,k_s)) \rightarrow m_p$. Таким чином розшифровується передостанній символ повідомлення, який використовує ключ. Всі наступні елементи шифрування будуть здійснюватися між символами самих повідомлень, а саме, наступний крок шифрування буде здійснюватися за таким набором:

$$c_1 c_2\; c_3\, c_4 \dots\, c_p \,||\, c_{p+1}$$

оскільки розшифрування здійснюється поелементно, тобто:

$$\leftarrow c_1 c_2\, c_3\, c_4 \dots\, c_p \,||\, c_{p+1}$$

згідно з II законом Шредера спочатку виконується операція $L(c_1,c_{p+1}) \rightarrow m_{p+1}^*$, ,а потім ця ж сама операція, але між іншими елементами, а саме $L(m_{p+1}^*,L(c_{p+}1,c1)) \rightarrow m_{p+1}$. Таким чином розшифровується останній символ.

Таким чином розшифрована послідовність згідно I закону Шредера має загальний вигляд:

$k_1c_1 \cdot\; c_1k_1$,$k_2c_2 \cdot\; c_2k_2$, $k_3c_3 \cdot\; c_3k_3$, $k_4c_4 \cdot\; c_4k_4$,…,$k_sc_k \cdot\; c_kk_s$,…, $c_1c_{p+1} \cdot\; c_{p+1}c_1$

Отже, множина розшифрованого повідомлення $M = \{\, m_1\, m_2\, m_3\, m_4\, \dots\, m_p\, \}$

2.4.1 Загальна схема розшифрування за квазігруповою тотожністю II-го закону Шредера

Вхідні дані:

- Множина символів зашифрованого повідомлення C={c_1 ,c_2 ,c_3 ,c_3,…,c_{p+1}}

- Множина ключів K={ k_1 k_2 k_3 k_4 … k_s }

Проміжні обчислення:

- L це таблиця Келі квазігрупи за I законом Шредера

- Множина проміжних шифрованих значень M={m_1^* , m , m_3^*, m_4^*,…, m_{p+1}^*}

Вихідні дані:

- Множина розшифрованого повідомлення M ={ m_1 m_2 m_3 m_4 … m_p }

$k_1\,k_2\,k_3\,k_4\ \ldots\ k_s \,||\, c_1\,c_2\,c_3\,c_4\ \ldots\ c_p$

$L(k_1,c_1) \rightarrow m_1^*$, $L(m_1^*, L(c_1,k_1)) \rightarrow m_1$

$\leftarrow k_1\,k_2\,k_3\,k_4\ \ldots\ k_s \,||\, c_1\,c_2\,c_3\,c_4\ \ldots\ c_p$

$k_2\,k_3\,k_4\ \ldots\ k_s\ \ c_1 || \, c_2\,c_3\,c_4\ \ldots\ c_p$

$L(k_2,c_2) \rightarrow m_2^*$, $L(m_2^*, L(c_2,k_2)) \rightarrow m_2$

$\leftarrow k_2\,k_3\,k_4\ \ldots\ k_s\ \ c_1 || \, c_2\,c_3\,c_4\ \ldots\ c_p$

$k_3\,k_4\ \ldots\ k_s\ \ c_1\,c_2 \,||\, c_3\,c_4\ \ldots\ c_p$

$L(k_3,c_3) \rightarrow m_3^*$, $L(m_3^*, L(c_3,k_3)) \rightarrow m_3$

$\leftarrow k_3\,k_4\ \ldots\ k_s\ \ c_1c_2 \,||\, c_3\,c_4\ \ldots\ c_p$

$k_4\ \ldots\ k_s\ \ c_1\,c_2c_3 \,||\, c_4\ \ldots\ c_p$

$L(k_4,c_4) \rightarrow m_4^*$, $L(m_4^*, L(k_4,c_4)) \rightarrow m_4$

$\leftarrow k_4\ \ldots\ k_s\,c_1\,c_2\,c_3 \,||\, c_4\ \ldots\ c_p$

…

$k_s\ \ c_1\,c_2\,c_3\,c_4 || \ \ldots\ c_p$

$L(k_s,c_p) \rightarrow m_p^*$, $L(m_p^*, L(c_p k_s)) \rightarrow m_p$

$\leftarrow k_s\,c_1c_2\,c_3c_4 \ ||\ \ldots\ c_p$

$c_1\,c_2\,c_3\,c_4\ \ldots\ c_p || \ c_{p+1}$

$L(c_1,c_{p+1}) \rightarrow m_{p+1}^*$, $L(m_{p+1}^*, L(c_{p+1},c_1)) \rightarrow m_{p+1}$

…

3 ЗАСІБ ШИФРУВАННЯ ДАННИХ

3.1 Розробка структури засобу шифрування за І законом Шредера

Шифрування за І законом Шредера базується на використанні квазігрупових операцій та псевдовипадкових послідовностей. В основі цього процесу лежать такі етапи:

1 Генерація псевдовипадкової послідовності для забезпечення ключової інформації.

2 Використання регістрів зсуву для маніпуляції повідомленням і ключовими даними.

3 Виконання операцій над даними в операційному блоці із застосуванням квазігруп.

4

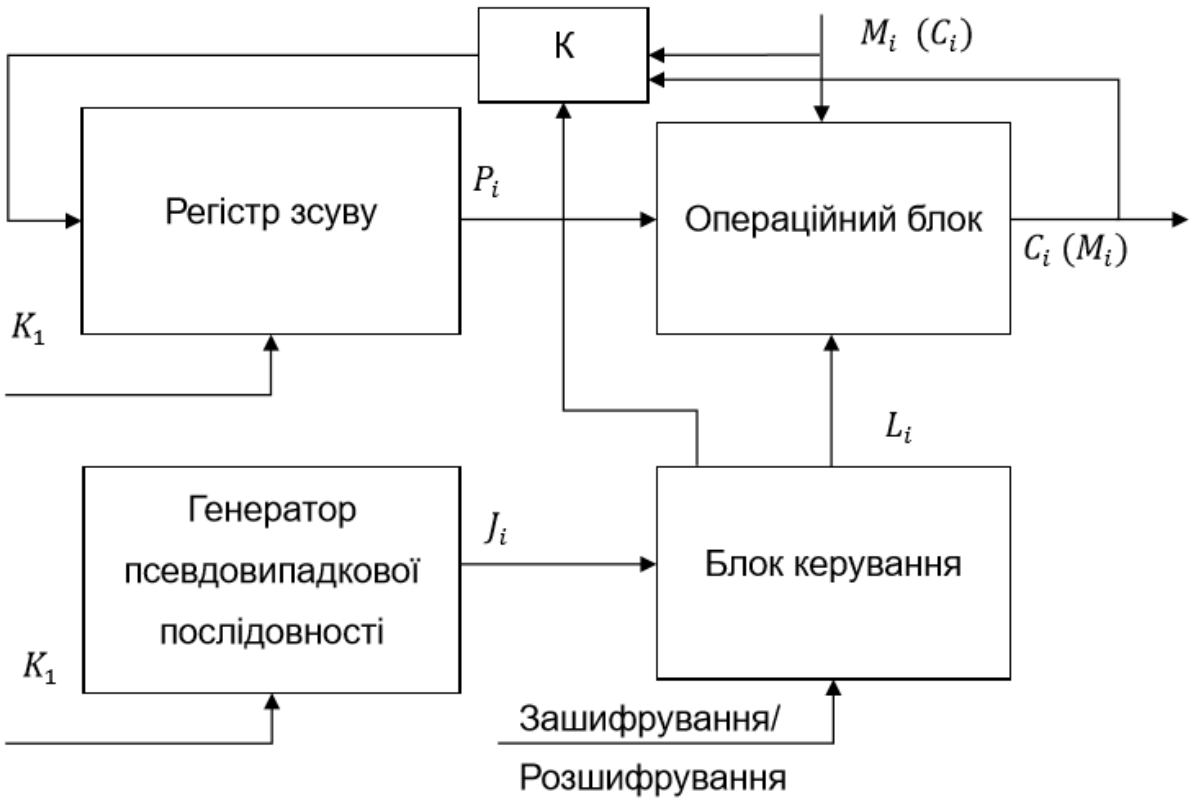


Рисунок 3.1 – Структура засобу шифрування

Структурні блоки апаратного засобу:

1. Генератор псевдовипадкової послідовності (ГПП)

- Основний компонент для створення ключів шифрування.
- Генерація 32-бітної послідовності на основі регістрів зсуву з лінійним зворотним зв’язком.
- Формує виходи J_i, які надходять до блоку керування для визначення активної квазігрупи.

2. Регістр зсуву

- Складається з 32 тригерів для маніпуляції з даними.
- Передає два біти ключа K_1в операційний блок для виконання логічних операцій.
- Здійснює циклічний і нециклічний зсуви, оновлюючи вміст комірок.

3. Комутатор (К)

- Координує потік даних між вхідними повідомленнями M_i, регістром зсуву та операційним блоком.
- Використовує керуючий сигнал із блоку керування для вибору режиму роботи (шифрування/розшифрування).

4. Блок керування

- Центральний вузол синхронізації апаратного засобу.
- Формує сигнали керування L_i для операційного блоку на основі дешифратора.
- Обирає активну квазігрупу для поточної операції, враховуючи сигнали J_i з ГПП.

5. Операційний блок

- Виконує логічні операції на основі основного й побічного набору квазігруп.
- Реалізує обчислення шифрованих даних $C_i(M_i)$ за допомогою кон’юнкції та диз’юнкції сигналів L_i, P_i.

Алгоритм роботи

Етап 1: Шифрування

1. Вхідне повідомлення M_i передається до комутатора.
2. Комутатор направляє дані до регістра зсуву.
3. Генератор псевдовипадкової послідовності формує ключі K_1, які регістр зсуву подає в операційний блок.
4. Блок керування визначає поточну квазігрупу через дешифратор і сигнал L_i.
5. Операційний блок виконує обчислення шифрованого повідомлення $C_i(M_i)$, яке подається на вихід.

Етап 2: Розшифрування

1. Шифроване повідомлення C_i надходить до комутатора.
2. Комутатор направляє дані в регістр зсуву для обробки.
3. Операційний блок, використовуючи сигнали L_i і ключі K_1, виконує обернені операції для відновлення початкового повідомлення M_i.

Схема апаратного засобу

Дана схема ідеально демонструє взаємодію між блоками:

- Регістр зсуву — центральний вузол для передачі ключових даних.
- Генератор псевдовипадкової послідовності — забезпечує криптографічну стійкість.
- Операційний блок — реалізує всі логічні операції на основі обраної квазігрупи.

Ця структура відображає апаратний засіб для реалізації І закону Шредера і є основою для ефективного шифрування.

3.2 Модуль побудови

Ця робота була виконана на мові програмування Python з використанням бібліотеки NumPy для роботи з матрицями. Модуль побудови криптографічного алгоритму на основі латинських квадратів є ключовим компонентом системи шифрування. Основною метою цього модуля є забезпечення безпечного перетворення числових даних шляхом використання

властивостей латинських квадратів, які гарантують унікальність елементів у кожному рядку та стовпці. У цьому розділі описується структура модуля, його алгоритми та програмна реалізація.

Модуль включає такі основні компоненти:

- Попередньо визначений латинський квадрат**:** матриця, яка використовується для перетворення чисел.
- Таблиця Келі**:** структура, яка визначає правила перетворення чисел.

Ключовим елементом є використання заздалегідь підготовленого латинського квадрата, який забезпечує уникнення повторень у кожному рядку та стовпці.

Попередньо визначений латинський квадрат

У цьому модулі використовується готовий латинський квадрат, представлений у вигляді матриці:

```
latin_square_from_image = [
 [0, 1, 2, 3, 4, 5, 6, 7],
 [1, 0, 3, 2, 5, 4, 7, 6],
 [2, 3, 1, 0, 6, 7, 5, 4],
 [3, 2, 0, 1, 7, 6, 4, 5],
 [4, 5, 6, 7, 0, 1, 2, 3],
 [5, 4, 7, 6, 1, 0, 3, 2],
 [6, 7, 5, 4, 2, 3, 1, 0],
  [7, 6, 4, 5, 3, 2, 0, 1]
]
```

Ця таблиця забезпечує унікальність значень у кожному рядку та стовпці, що є необхідним для коректного функціонування системи.

Модуль побудови криптографічного алгоритму на основі латинських квадратів забезпечує ефективну генерацію структур, що можуть бути використані для подальших криптографічних перетворень. Унікальність елементів у рядках і стовпцях латинського квадрата гарантує коректність побудови таблиці Келі, що є фундаментом для криптографічних операцій.

3.3 Модуль зашифрування

Модуль зашифрування відповідає за перетворення відкритого тексту у зашифроване повідомлення, використовуючи латинський квадрат і динамічне оновлення ключів. Його основною метою є забезпечення стійкого до атак перетворення на основі подвійного зсуву символів.

Процес розпочинається з ініціалізації вхідних даних. У функції `encrypt_with_shift` створюються копії повідомлення та ключів для уникнення змін оригінальних списків. Порожній список `encrypted_message` слугує для збереження результатів шифрування. Цей етап реалізований наступним чином:

```
encrypted_message = []
temp_message = message.copy()
temp_keys = keys.copy()
```

Далі запускається цикл, який обробляє кожен символ повідомлення. На початку ітерації визначається поточний символ mi та ключ key. Ці значення використовуються для зсувів у латинському квадраті:

```
for _ in range(len(message)):
    key = temp_keys[0]
    mi = temp_message[0]
```

Перший зсув виконується за допомогою латинського квадрата, де використовується рядок, що відповідає поточному ключу, та стовпець, що відповідає символу повідомлення. Значення зчитується за формулою `latin_square[key][mi]` і зберігається у змінній `intermediate`:

```
intermediate = latin_square[key][mi]
```

Після цього виконується другий зсув. Для цього використовується значення `intermediate`, яке тепер слугує індексом рядка в латинському квадраті.

У поєднанні з тим самим символом `mi` обчислюється фінальне зашифроване значення `cipher`:

```
cipher = latin_square[intermediate][mi]
```

Отримане значення додається до списку `encrypted_message`, формуючи зашифроване повідомлення:

```
encrypted_message.append(cipher)
```

Щоб створити динамічність алгоритму, ключі зсуваються циклічно. Перший ключ переміщується в кінець списку. Це гарантує, що кожен символ обробляється унікальним чином:

```
temp_keys = temp_keys[1:] + [key]
```

Аналогічно зсувається і повідомлення: перший символ переміщується у кінець списку, щоб послідовність обробки символів змінювалась на кожній ітерації:

```
temp_message = temp_message[1:] + [mi]
```

Коли всі символи повідомлення зашифровані, функція повертає список `encrypted_message`, що містить зашифровані числові значення:

```
return encrypted_message
```

Цей алгоритм забезпечує надійне шифрування даних, використовуючи два рівні зсуву та динамічне оновлення ключів і повідомлення. Це ускладнює аналіз зашифрованого тексту і гарантує високу криптостійкість.

3.4 Модуль розшифрування

Модуль розшифрування реалізує процес відновлення вихідного тексту із зашифрованого повідомлення, використовуючи ті ж ключі та латинський квадрат, які були застосовані під час шифрування. Головною метою є виконання зворотних операцій для кожного символу зашифрованого тексту, відновлюючи початкове повідомлення.

Процес розшифрування починається з ініціалізації вхідних даних. У функції `decrypt_with_shift` створюються копії списків ключів для їх циклічного оновлення, а також порожній список для збереження результатів розшифрування:

```
decrypted_message = []
temp_keys = keys.copy()
```

Далі для кожного зашифрованого символу запускається цикл. Поточний ключ (key) використовується для визначення рядка у латинському квадраті, а зашифрований символ (ci) — для виконання зворотного пошуку. Ці значення ініціалізуються на кожній ітерації:

```
for ci in encrypted_message:
    key = temp_keys[0]
```

Для відновлення початкового символу виконується подвійний зворотний пошук у латинському квадраті. Спочатку перебираються всі можливі значення у рядку, який відповідає поточному ключу (key). Якщо значення проміжного зсуву (intermediate) задовольняє умову, його використовують для наступного етапу:

```
for x in range(len(latin_square)):
  intermediate = latin_square[key][x]
if latin_square[intermediate][x] == ci:
```

```
decrypted_message.append(x)
break
```

Після успішного пошуку символу в латинському квадраті оновлюється список ключів, щоб забезпечити циклічне використання. Поточний ключ переміщується в кінець списку, зберігаючи порядок зсувів, аналогічний шифруванню:

```
temp_keys = temp_keys[1:] + [key]
```

Коли всі символи зашифрованого повідомлення оброблено, функція повертає список розшифрованих числових значень:

```
return decrypted_message
```

Розшифрування базується на зворотному виконанні операцій шифрування.

Модуль розшифрування є невід’ємною частиною системи шифрування, яка забезпечує можливість точного відновлення текстових даних із зашифрованого формату.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Комерційний та технологічний аудит проводиться з метою розробки криптографічного методу шифрування, що базується на квазігрупових тотожностях, з метою забезпечення стійкості до сучасних криптоаналітичних атак та підвищення ефективності алгоритму для застосування в системах захисту інформації.

Для проведення технологічного аудиту було залучено 3-х незалежних експертів Вінницького національного технічного університету, кафедри захисту інформації: к.т.н., доцент Леонід Куперштейн, д. т. н., професор Володимир Лужецький, к. фіз.-мат. н. Галина Шелепало.

Для проведення технологічного аудиту було використано таблицю 4.1 в якій за п’ятибальною шкалою використовуючи 12 критеріїв здійснено оцінку комерційного потенціалу.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Продовження табл. 4.1

5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Таблиця 4.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

В таблиці 4.3 наведено результати оцінювання експертами комерційного потенціалу розробки.

Таблиця 4.4 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	доц. кафедри ЗІ, к.т.н., доцент Леонід КУПЕРШТЕЙН	Зав. кафедри ЗІ, д. т. н., професор Володимир ЛУЖЕЦЬКИЙ	к. фіз.-мат. н., доцент каф. ЗІ Галина ШЕЛЕПАЛО
	Бали, виставлені експертами:		
1	4	4	4
2	3	3	3
3	3	3	4
4	4	4	4
5	4	4	4
6	4	4	4
7	4	4	4
8	4	4	4
9	3	3	3
10	4	4	4
11	4	4	4
12	3	3	3
Сума балів	$СБ_1$=31	$СБ_2$=34	$СБ_3$=34
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{1}^{3} СБ_i}{3} = \frac{31+34+34}{3} = 33$		

Середньоарифметична оцінка, отримана на основі експертних висновків,

становить 33 бали, і згідно з таблицею 4.3, це вказує на вище середнього рівень комерційного потенціалу результатів проведених досліджень.

Результатом роботи є створений криптографічний метод, заснований на квазігрупових тотожностях, а також програмна реалізація алгоритмів шифрування та дешифрування даних. Розроблений метод може бути використаний підприємствами, що займаються захистом інформації, банківськими структурами, державними органами, а також у корпоративних системах для захисту конфіденційних даних.

В якості аналога для розробки було обрано алгоритм шифрування AES (Advanced Encryption Standard), який є одним із найбільш поширених методів симетричного шифрування.

Основними недоліками аналога є:

1. Високі обчислювальні витрати при реалізації на пристроях із обмеженими ресурсами, таких як IoT-пристрої.
2. Уразливість до атак за побічними каналами, наприклад, аналіз споживаної енергії або часу виконання операцій.
3. Обмежена гнучкість у налаштуванні під конкретні вимоги користувача (наприклад, нестандартні розміри блоків або ключів).

Також до недоліків можна віднести складність впровадження для малих і середніх підприємств через необхідність використання додаткового апаратного або програмного забезпечення для підвищення стійкості до атак.

У розробці дана проблема вирішується шляхом використання квазігрупових тотожностей, які дозволяють створювати більш легкі у реалізації та адаптивні криптографічні алгоритми. Запропонований метод забезпечує високу швидкість шифрування, низькі вимоги до ресурсів та стійкість до атак на основі статистичного аналізу.

Аналіз існуючих методів вирішення задачі показав, що більшість сучасних алгоритмів шифрування (RSA, DES, AES) мають високу криптографічну стійкість, але вимагають значних обчислювальних ресурсів, що ускладнює їх використання в умовах обмежень. Крім того, багато з цих

алгоритмів не враховують специфіку захисту невеликих обсягів даних або потребу в адаптації до нестандартних систем.

Також система випереджає аналог за такими параметрами, як:

1. Швидкість шифрування для малих повідомлень.
2. Простота інтеграції в системи з обмеженими ресурсами.
3. Гнучкість у налаштуванні алгоритму під специфічні потреби користувача.
4. Зменшення вразливості до атак за побічними каналами завдяки використанню квазігрупових таблиць.

Запропонована розробка може стати конкурентоспроможним рішенням для забезпечення безпеки даних в умовах зростаючих вимог до адаптивності та економічності криптографічних методів.

4.2 Прогнозування витрат на виконання науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи групуються за такими статтями: витрати на оплату праці, витрати на соціальні заходи, матеріали, паливо та енергія для науково-виробничих цілей, витрати на службові відрядження, програмне забезпечення для наукових робіт, інші витрати, накладні витрати.

1. Основна заробітна плата кожного із дослідників $З_О$, якщо вони працюють в наукових установах бюджетної сфери визначається за формулою:

$$З_0 = \frac{М}{Т_р} * t \text{ (грн)} \quad (4.1)$$

де $М$ – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

$Т_Р$ – число робочих днів в місяці; приблизно $Т_Р \approx 21...23$ дні;

t – число робочих днів роботи дослідника.

Зведемо сумарні розрахунки до таблиця 4.5.

Таблиця 4.5 – Заробітна плата дослідника в науковій установі бюджетної сфери

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату грн.
Керівник	20000	952,4	5	4762
Програміст	19000	904,8	21	19000
Всього				23762

2. Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт розраховують за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i, \quad (4.2)$$

де $С_і$ – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

$t_і$ – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду $С_і$ можна визначити за формулою:

$$C_i = \frac{М_м \cdot К_і \cdot К_с}{Т_р \cdot t_{зм}}, \quad (4.3)$$

де $М_М$ – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), грн;

$К_і$ – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду;

$К_с$ – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих

об’єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$Т_р$ – середня кількість робочих днів в місяці, приблизно Тр = 21…23 дні;

$t_{зм}$ – тривалість зміни, год.

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Погодинна тарифна ставка, грн	Величина оплати на робітника, грн
1. Розробка алгоритму	2	2	52,4	104,8
2. Написання коду	5	3	64,3	321,4
3. Налагодження програми	2	5	81,0	161,9
4. Випробувальні	8	2	52,4	419,0
5. Документування системи	3	2	71,4	214,3
Всього				1221,4

3. Розрахунок додаткової заробітної плати робітників

Додаткова заробітна плата $З_д$ всіх розробників та робітників, які приймали устають в розробці нового технічного рішення розраховується як 10 - 12 % від основної заробітної плати робітників.

На даному підприємстві додаткова заробітна плата начисляється в розмірі 11% від основної заробітної плати.

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

$$З_д = 0{,}11 * (23762 + 1221{,}4) = 2748{,}17 \text{ (грн)}$$

4. Нарахування на заробітну плату $Н_{ЗП}$ дослідників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою (4.10):

$$Н_{ЗП} = (З_о + З_р + З_д) * \frac{\beta}{100} \text{ (грн)} \quad (4.3)$$

де $З_о$ – основна заробітна плата розробників, грн.;

$З_д$ – додаткова заробітна плата всіх розробників та робітників, грн.;

$З_р$ – основну заробітну плату робітників, грн.;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % .

Дана діяльність відноситься до бюджетної сфери, тому ставка єдиного внеску на загальнообов'язкове державне соціальне страхування буде складати 22%, тоді**:**

$$Н_{ЗП} = (23762 + 1221{,}4 + 2748{,}17) * \frac{22}{100} = 6100{,}93 \text{ (грн)}$$

5. Сировина та матеріали.

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби й предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за прямим призначенням згідно з нормами їх витрачання, а також витрачені придбані напівфабрикати, що підлягають монтажу або виготовленню й додатковій обробці в цій організації, чи дослідні зразки, що виготовляються виробниками за документацією наукової організації.

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

$$М = \sum_{i=1}^{n} H_j \cdot Ц_j \cdot К_j - \sum_{i=1}^{n} В_j \cdot Ц_{вj}, \qquad (4.5)$$

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

n – кількість видів матеріалів;

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

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

Вj – маса відходів j-го найменування, кг;

Цвj – вартість відходів j-го найменування, грн/кг.

Проведені розрахунки зведені в таблицю 4.9.

Таблиця 4.7 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, шт	Вартість витраченого матеріалу, грн
Папір	180	1	180
Ручка	20	1	20
Блокнот	50	1	50
Флешка	300	1	300
З врахуванням коефіцієнта транспортування			605

6. Програмне забезпечення для наукових (експериментальних) робіт

Балансову вартість програмного забезпечення розраховують за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (4.6)$$

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

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

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1,10…1,12$);

k – кількість найменувань програмних засобів.

Отримані результати необхідно звести до таблиці:

Таблиця 4.8 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
MATLAB (Academic License)	1	25000	25000
Python IDE (PyCharm Professional)	1	5000	5000
Microsoft Visual Studio	1	15000	15000
Інструменти для моделювання (Simulink)	1	10000	10000

Система тестування (TestRail)	1	8000	8000
Всього з врахуванням налагодження			63000

7. Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.7)$$

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

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

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

Проведені розрахунки необхідно звести до таблиці 4.9.

Таблиця 4.9 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Комп'ютер	40000	2	1	1666,67
Всього				1666,67

8. До статті «Паливо та енергія для науково-виробничих цілей» відносяться витрати на всі види палива й енергії, що безпосередньо використовуються з технологічною метою на проведення досліджень.

$$В_е = \sum_{i=1}^{n} \frac{W_{yt} \cdot t_i \cdot Ц_е \cdot К_{впі}}{\eta_i} \quad (4.8)$$

де W_{yt} – встановлена потужність обладнання на певному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_е$ – вартість 1 кВт-години електроенергії, грн;

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

$\eta_і$ – коефіцієнт корисної дії обладнання, $\eta_і < 1$.

Для написання магістерської роботи використовується персональний комп'ютер для якого розрахуємо витрати на електроенергію.

$$В_е = \frac{0{,}5 \cdot 160 \cdot 10{,}5 \cdot 0{,}5}{0{,}8} = 525$$

9. Службові відрядження.

Витрати за статтею «Службові відрядження» розраховуються як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{св} = (З_о + З_р) * \frac{Н_{св}}{100\%}, \quad (4.9)$$

де $Н_{св}$ – норма нарахування за статтею «Службові відрядження».

$$В_{св} = 0{,}2 * (23762 + 1221{,}4) = 4996{,}67$$

10. Накладні (загальновиробничі) витрати Внзв охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо. Накладні (загальновиробничі) витрати Внзв можна прийняти як (100…150)% від суми основної заробітної плати розробників та робітників, які виконували дану МКНР, тобто:

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.10)$$

де $Н_{нзв}$ – норма нарахування за статтею «Інші витрати».

$$В_{нзв} = (23762 + 1221{,}4) \cdot \frac{100}{100\%} = 24983{,}33 \text{ грн}$$

Сума всіх попередніх статей витрат дає витрати, які безпосередньо стосуються даного розділу МКНР

В=23762+1221,4+2748,17+6100,93+605+63000+1666,67+525+4996,67+24983,

ЗЗ=129609,09 грн

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

$$ЗВ = \frac{В}{\eta}, \quad (4.11)$$

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

Оскільки, робота знаходиться на стадії науково-дослідних робіт, то коефіцієнт β = 0,7.

Звідси:

$$ЗВ = \frac{129609,09}{0,7} = 185155,85 \text{ грн.}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки

У даному підрозділі кількісно спрогнозуємо, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної наукової роботи. Розрахуємо збільшення чистого прибутку підприємства $\Delta П_i$, для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, за формулою

$$\Delta П_i = \sum_{1}^{n} \left(\Delta Ц_o \cdot N + Ц_o \cdot \Delta N \right)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\upsilon}{100} \right) \quad (4.12)$$

де $\Delta Ц_0$ – покращення основного оціночного показника від впровадження результатів розробки у даному році.

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

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки:

$Ц_o$ – основний оціночний показник, який визначає діяльність підприємства у

даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки:

л – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт *л* = 0,8333.

р – коефіцієнт, який враховує рентабельність продукту. р = 0,25;

х – ставка податку на прибуток. У 2024 році – 18%.

Припустимо, що ціна зросте на 500 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року на 800 шт., протягом другого року – на 900 шт., протягом третього року на 1000 шт. Реалізація продукції до впровадження розробки складала 1 шт., а її ціна до 2000 грн. Розрахуємо прибуток, яке отримає підприємство протягом трьох років.

$$\Delta П_1 = [500 \cdot 1 + (2000 + 500) \cdot 800] \cdot 0{,}833 \cdot 0{,}25 \cdot \left(1 + \frac{18}{100}\right) = 341738{,}41 \text{ грн.}$$

$$\Delta П_2 = [500 \cdot 1 + (2000 + 500) \cdot (800 + 900)] \cdot 0{,}833 \cdot 0{,}25 \cdot \left(1 + \frac{18}{100}\right) = 726512{,}63 \text{ грн.}$$

$$\Delta П_3 = [500 \cdot 1 + (2000 + 500) \cdot (800 + 900 + 1000)] \cdot 0{,}833 \cdot 0{,}25 \cdot \left(1 + \frac{18}{100}\right) = 1153578{,}9 \text{грн.}$$

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Розрахуємо основні показники, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахуємо величину початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки.

$$PV = k_{\text{інв}} \cdot \text{ЗВ}, \qquad (4.13)$$

$k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо ($k_{\text{інв}} = 2…5$).

$$PV = 2 \cdot 185155{,}85 \quad = 370311{,}7$$

Розрахуємо абсолютну ефективність вкладених інвестицій $E_{абс}$ згідно наступної формули:

$$E_{абс} = (ПП - PV) \qquad (4.14)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство від реалізації результатів наукової розробки, грн.;

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

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДЦКР, грн.;

Т – період часу, протягом якою виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,2;

t – період часу (в роках).

$$\text{ПП} = \frac{341738{,}41}{(1+0{,}2)^1} + \frac{726512{,}63}{(1+0{,}2)^2} + \frac{1153578{,}9}{(1+0{,}2)^3} = 1459990{,}06 \text{ грн.}$$

$$\text{Е}_{\text{абс}} = (1459990{,}06 \ - 370311{,}7) = 1089678{,}36 \text{ грн.}$$

Оскільки $E_{абс} > 0$ то вкладання коштів на виконання та впровадження результатів НДДКР може бути доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього користуються формулою:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \tag{4.16}$$

$T_ж$ – життєвий цикл наукової розробки, роки.

$$\text{Е}_\text{в} = \sqrt[3]{1 + \frac{1089678{,}36}{370311{,}7}} - 1 = 0{,}90 = 90\%$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \tag{4.17}$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні d= (0,14…0,2);

f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,1).

$$\tau_{\min} = 0{,}18 + 0{,}05 = 0{,}23.$$

Так як $E_в > \tau_{\min}$ то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_в} \tag{4.18}$$

$$\text{Т}_\text{ок} = \frac{1}{0{,}9} = 1{,}1 \text{ роки}$$

Так як $T_{ок} \leq$ 3...5-ти років, то фінансування даної наукової розробки в принципі є доцільним.

4.5 Висновки до економічного розділу

Результати здійсненого технологічного аудиту вказують на вище середнього рівень комерційного потенціалу. У порівнянні з аналогічним виробом виявлено, що нова розробка вищої якості і більш конкурентоспроможна, як з технічних, так і економічних позначень.

Вкладені інвестиції в даний проект окупляться через 1,1 роки. Загальні витрати складають 185155,85 грн.

ВИСНОВКИ

У магістерській кваліфікаційній роботі виконано теоретичне обґрунтування та описано розробку методу шифрування на основі квазігрупових тотожностей, що дозволяє створювати алгоритми з високим рівнем криптографічної стійкості. У роботі доведено, що квазігрупові алгебри, задані тотожностями, завдяки своїм властивостям, є потужним інструментом для побудови криптографічних алгоритмів, здатних забезпечувати захист даних навіть в умовах активного розвитку методів криптоаналізу.

В ході роботи проведено огляд і аналіз сучасних криптографічних методів, зокрема тих, що базуються на групах, кільцях та квазігрупах. Крім того розроблено метод шифрування на основі квазігрупової тотожності, що описує многовид квазігруп закону Шредера, описано алгоритм шифрування та заштфрування за законом Шредера,а також розроблено структуру засобу з метою покращення рівня криптографічної стійкості.

Практична реалізація алгоритму виконана мовою Python із використанням сучасних бібліотек для роботи з алгебричними структурами. Це дозволило не лише продемонструвати працездатність запропонованого підходу, але й забезпечити його доступність для подальшого використання та інтеграції в інформаційні системи. В ході тестування алгоритму було підтверджено, що його рівень відповідає сучасним вимогам до ефективності та безпеки, оскільки реалізує високошвидкісні операції шифрування й розшифрування з мінімальними вимогами до апаратних ресурсів.

Таким чином, магістерська кваліфікаційна робота досягла своєї мети — створено метод і засіб шифрування, на основі квазігрупових тотожностей. Отримані результати мають теоретичну і практичну значущість, демонструють високу якість виконання і можуть бути основою для подальших досліджень у цій сфері.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Шелепало Г. В. Класифікація квазігрупових функційних рівнянь і тотожностей занімальної довжини, дис.... канд. фіз-мат. наук: 01.01.06. Хмельницьк, 2018. 27-28 с.
2. Evans . Abstract or values. Dale Machatice Joonial 30(21/33-317)
3. .Bakhtiari, S., Safavi-Naini, R., Pieprzyk, J.: A Message Authentication Code based on Latin Square. In V. Varadharajan, J. Pieprzyk, and Y. Mu (Eds.) Proceedings of ACISP'97, LNCS 1270, pp. 194–203), 1997. Springer Berlin Heidelberg.
4. Meyer, K. A.: A new message authentication code based on the non-associativity of quasi-groups. Doctoral dissertation. Iowa State University, 2006.
5. Gligoroski, D., Markovski, S., Knapskog, S. J.: The Stream Cipher Edon80. In: New Stream Cipher Designs: The eSTREAM Finalists, LNCS 4986, pp. 152–169, 2008. Springer-Verlag
6. CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness, 2013, https://competitions.cr.yp.to/caesar.html. Cited 19 Aug 2022
7. Bertoni, G., Daemen, J., Peeters, M., and Assche, G.: Duplexing the Sponge: Single-Pass Authenticated Encryption and Other Applications. In Selected Areas in Cryptography: 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers, pp. 320–337, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
8. Gligoroski, D, Mihajloska, H, Otte, D.: GAGE and InGAGE v1.0. Submission to the NIST's Lightweight Cryptography Standardization Process, 2019.
9. NIST Lightweight Cryptography Standardization Process, 2019 https://csrc.nist.gov/Projects/ lightweight-cryptography. Cited 19 Aug 2022.
10. Gligoroski, D., Markovski, S., Knapskog, S.J.: Multivariate Quadratic Trapdoor Functions based on Multivariate Quadratic Quasigroups. In: Proceedings of the American Conference on Applied Mathematics. MATH, World Scientific and Engineering Academy and Society (WSEAS), pp. 4449, 2008.

11. Gligoroski, D., Odegard, R.S., Jensen, R.E., Perret, L., Faugere, J.C., Knapskog, S.J., Markovski, S.: MQQ-SIG - An Ultra-Fast and Provably CMA Resistant
12. Markovski, S., Dimitrova, V., Trajcheska, Z., Petkovska, M., Kostadinoski, M., Buhov, D.: Block Cipher Defined by Matrix Presentation of Quasigroups. Proceedings of the 11th Conference on Informatics and Information Technology, CIIT 2014, Bitola
13. Markovski, S., Gligoroski, D., & Bakeva, V. (1999). Quasigroup string processing (pp. 13-28). Republic of Macedonia
14. Kumar, U., & Venkaiah, V. (2022). A Family of Block Ciphers Based on Multiple Quasigroups (p. 5). Retreived from https://eprint.iacr.org/2022/1761.
15. Gligoroski, D. (2001), Stream cipher based on quasigroup string transformation (p. 12). Faculty of Natural Sciences, Institute of Informatics.
16. Shcherbacov V. Elements of quasigroup theory. Elements of quasigroup theory and applications. 2017. P. 3–106. URL: https://doi.org/10.1201/9781315120058-2
17. Fawaz M., Zorkta H., Alnazer S. A public key cipher algorithm based on multivariate cubic quasigroups (MCQ). The international conference on electrical engineering. 2010. T. 7, № 7. C. 1–11. URL: https://doi.org/10.21608/iceeng.2010.33467
18. Gligoroski D. Stream cipher based on quasigroup string transformations. м. Skopje, 22 квіт. 2004 p. Skopje, 2004. C. 7–12.
19. Gligoroski D., Markovski S., Knapskog S.J., New stream cipher designs / ред.: M. Robshaw, O. Billet. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008. URL: https://doi.org/10.1007/978-3-540-68351-3.
20. Bjørstad T. E., A note on the Edon 80 s-box | Semantic Scholar. Semantic Scholar | AI-Powered Research Tool. URL: https://www.semanticscholar.org/paper/A-noteon-the-Edon-80-S-box-Bjørstad/fe7114a2ae9c605e026a5cfd61758f3d303f26f6

ДОДАТКИ

89

Додаток А

ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: Метод та засіб шифрування на основі квазігрупових тотожностей
Автор роботи: Качай Роман Володимирович

Тип роботи: магістерська кваліфікаційна робота
Підрозділ кафедра захисту інформації ФПКІ
Керівник Шелепало Галина Василівна, к.ф.-м.н., доцент .

Показники звіту подібності

Turnitin	
Оригінальність	91 %
Загальна схожість	9 %

Аналіз звіту подібності (відмітити потрібне):

√ 1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

□ 2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

□ 3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений з повним звітом подібності, який був згенерований Системою щодо роботи.

Автор роботи (підпис) Роман КАЧАЙ (прізвище, ініціали)

Особа, відповідальна за перевірку (підпис) Валентина КАПЛУН (прізвище, ініціали)

Керівник роботи (підпис) Галина ШЕЛЕПАЛО (прізвище, ініціали)

Додаток Б

Код програми

```
import numpy as np
latin_square_from_image = [
    [0, 1, 2, 3, 4, 5, 6, 7],
    [1, 0, 3, 2, 5, 4, 7, 6],
    [2, 3, 1, 0, 6, 7, 5, 4],
    [3, 2, 0, 1, 7, 6, 4, 5],
    [4, 5, 6, 7, 0, 1, 2, 3],
    [5, 4, 7, 6, 1, 0, 3, 2],
    [6, 7, 5, 4, 2, 3, 1, 0],
    [7, 6, 4, 5, 3, 2, 0, 1]
]

def generate_latin_square(order=8):
    return np.array(latin_square_from_image)

def encrypt_with_shift(message, keys, latin_square):
    encrypted_message = []
    temp_message = message.copy()
    temp_keys = keys.copy()
    for _ in range(len(message)):
        key = temp_keys[0]
        mi = temp_message[0]
        intermediate = latin_square[key][mi]
        cipher = latin_square[intermediate][mi]
        encrypted_message.append(cipher)
        temp_keys = temp_keys[1:] + [key]
        temp_message = temp_message[1:] + [mi]
    return encrypted_message

def decrypt_with_shift(encrypted_message, keys, latin_square):
    decrypted_message = []
    temp_keys = keys.copy()
    for ci in encrypted_message:
        key = temp_keys[0]
        for x in range(len(latin_square)):
```

```
            intermediate = latin_square[key][x]
            if latin_square[intermediate][x] == ci:
                decrypted_message.append(x)
                break
        temp_keys = temp_keys[1:] + [key]
    return decrypted_message

def main():
    order = 8
    latin_square = generate_latin_square(order)
    numeric_messages = [[1, 2, 3, 4], [4, 3, 2, 1], [0, 7, 6, 5], [7, 0,
1, 2]]
    keys = [3, 5, 7, 2]
    print("Оригінальні числові повідомлення:", numeric_messages)
    encrypted_messages = [encrypt_with_shift(message, keys, latin_square)
for message in numeric_messages]
    print("Зашифровані повідомлення:", encrypted_messages)
    decrypted_messages = [decrypt_with_shift(encrypted, keys,
latin_square) for encrypted in encrypted_messages]
    print("Розшифровані повідомлення:", decrypted_messages)
if __name__ == "__main__":
    main()
```

92

Додаток В

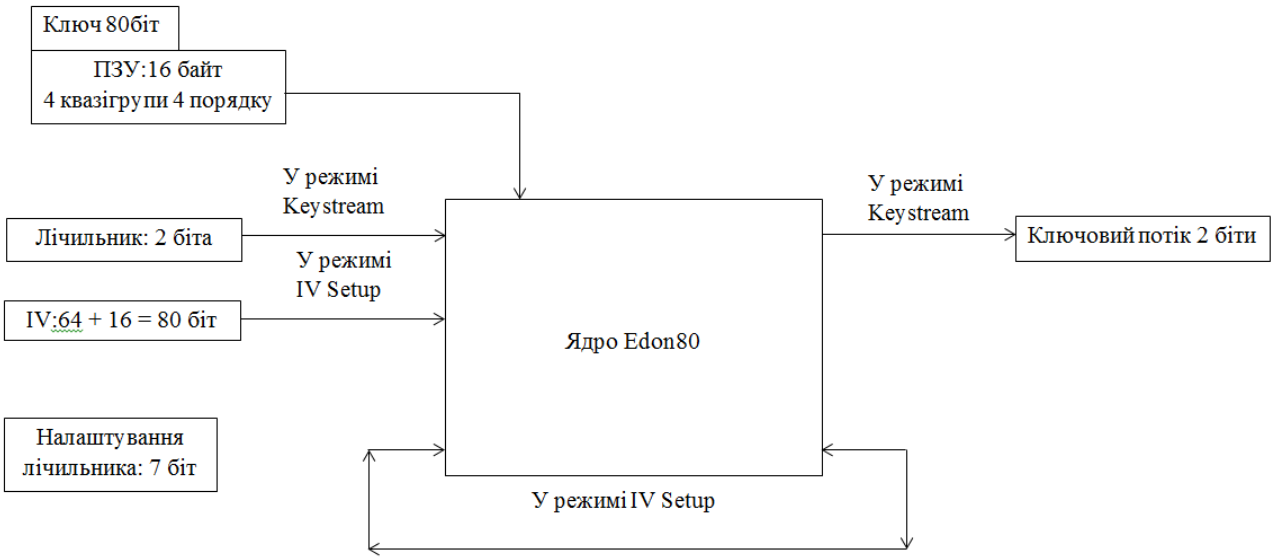
ІЛЮСТРАТИВНА ЧАСТИНА

«Метод та засіб шифрування на основі квазігрупових тотожностей»

Виконав студент 2-го курсу, групи
1БС-23м
спеціальності 125 - Кібербезпека
та захист інформації
_____ Роман КАЧАЙ
17. 12 2024 р.

Керівник: доц. кафедри ЗІ, к.фіз-мат.н.,
_____ Галина ШЕЛЕПАЛО
17. 12 2024 р.

Компоненти Edon80 та їх співвідношення



Квазігрупи, використані в дизайні Edon80

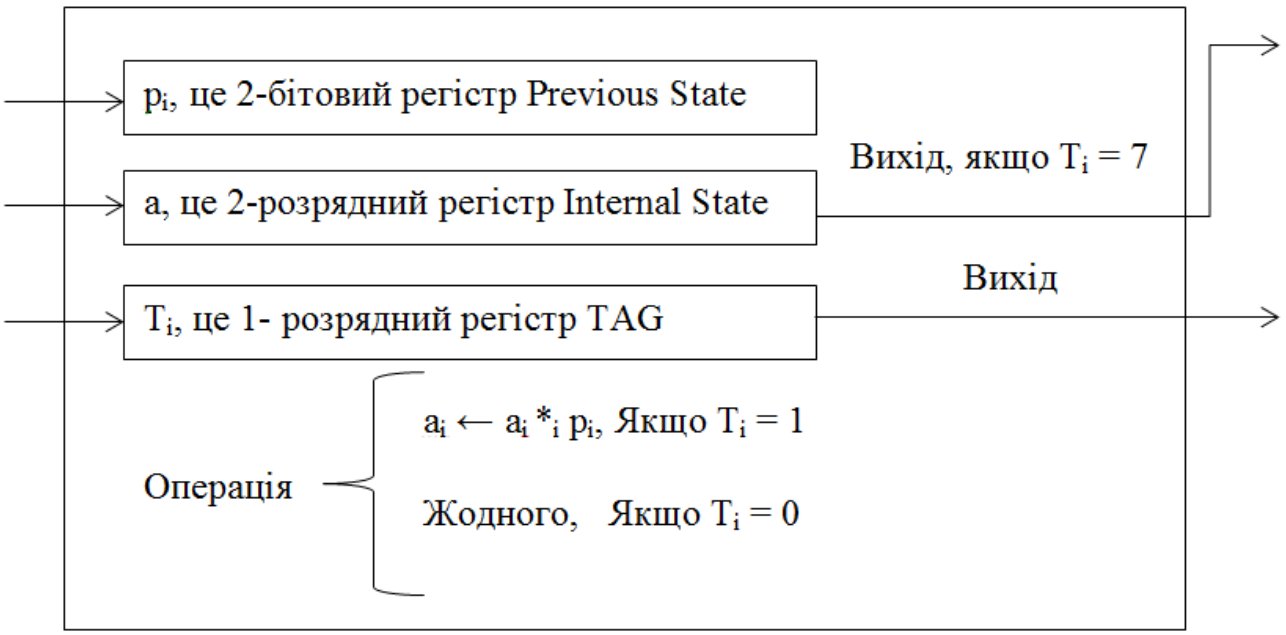
$\bullet_0$	0	1	2	3
0	0	2	1	3
1	2	1	3	0
2	1	3	0	2
3	3	0	2	1

$\bullet_1$	0	1	2	3
0	1	3	0	2
1	0	1	2	3
2	2	0	3	1
3	3	2	1	0

$\bullet_2$	0	1	2	3
0	2	1	0	3
1	1	2	3	0
2	3	0	2	1
3	0	3	1	2

$\bullet_3$	0	1	2	3
0	3	2	1	0
1	1	0	3	2
2	0	3	2	1
3	2	1	0	3

Схематичне зображення квазігрупової операції Edon80



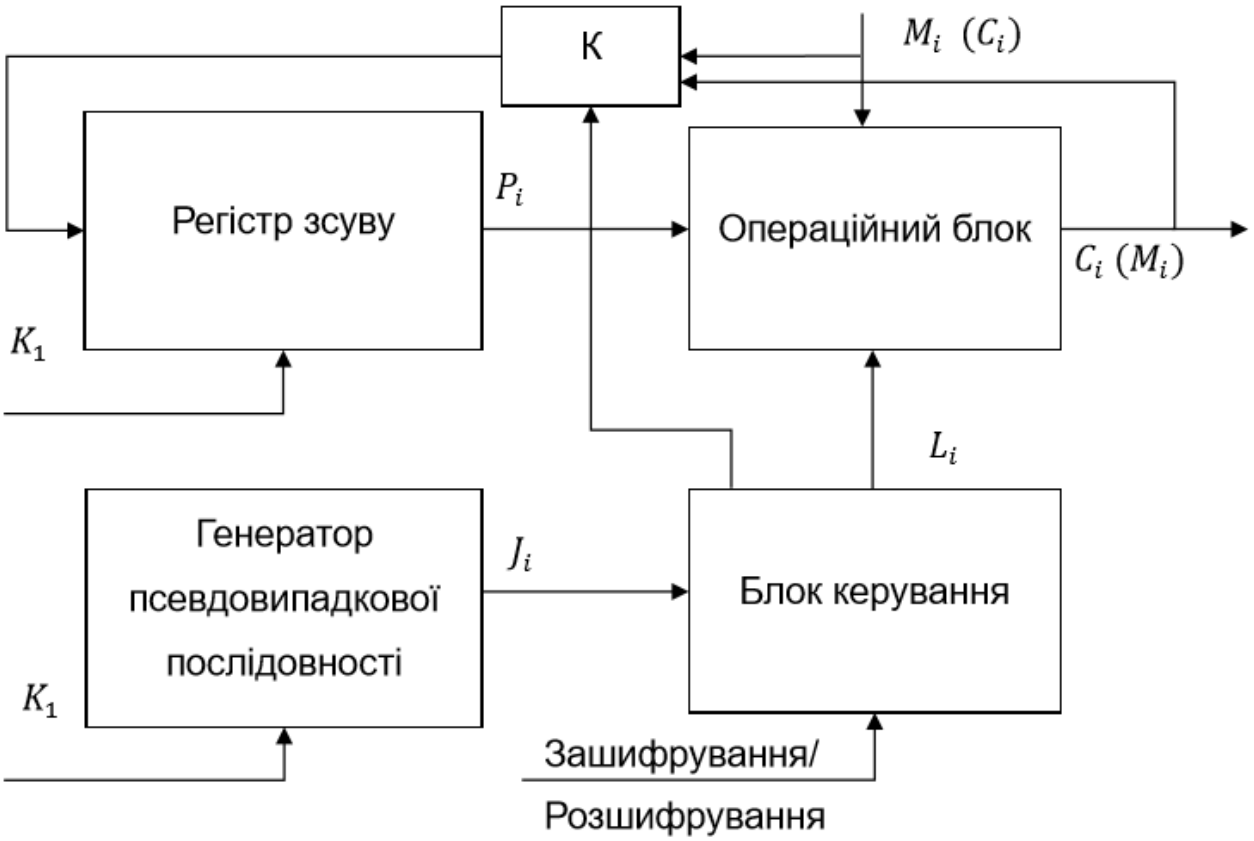
Довільний латинський квадрат 8-го порядку

*	0	1	2	3	4	5	6	7
0	1	0	3	2	4	5	6	7
1	0	2	1	3	5	6	7	4
2	3	1	2	0	6	7	4	5
3	2	3	0	1	7	4	5	6
4	4	5	6	7	0	1	2	3
5	5	6	7	4	1	2	3	0
6	6	7	4	5	2	3	0	1
7	7	4	5	6	3	0	1	2

Нормалізований латинський квадрат 8-го порядку

*	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	3	0	2	5	6	7	4
2	2	0	3	1	6	7	4	5
3	3	2	1	0	7	4	5	6
4	4	5	6	7	1	0	3	2
5	5	6	7	4	0	3	2	1
6	6	7	4	5	3	2	1	0
7	7	4	5	6	2	1	0	3

Структура засобу шифрування



ВІДГУК ОПОНЕНТА
на магістерську кваліфікаційну роботу

студента Качая Романа Володимировича
(прізвище, ім'я, по батькові)
на тему Метод та засіб шифрування на основі квазігрупових тотожностей

Подана магістерська кваліфікаційна робота оригінальна своєю ідеєю для розробки методу та засобу шифрування на основі інноваційних математичних структур, а саме квазігрупових тотожностей. Такі алгебричні структури, як квазігрупи та їх комбінаторні аналоги - латинські квадрати на сьогодні швидко розвиваються і мають застосування в багатьох галузях, особливо і криптографії.

Мета роботи полягає у підвищенні рівня захисту інформації за допомогою квазігрупової тотожності, що відповідає закону Шередера. Кожна операція, що фігурує в тотожності, є тотально-симетричною, тобто таблиця Келі одна й та сама для всіх оборотних операцій квазігрупи, а це, в свою чергу, дозволить зменшити навантаження на апаратну частину.

У розділі 1 проведено огляд та аналіз існуючих криптографічних методів і систем, розглянуто квазігрупові структури та їх комбінаторні аналоги.

У розділі 2 запропоновано алгоритм шифрування на квазігрупових тотожностях, що описують многовид квазігруп за законом Шредера. Наведено математичну модель, досліджено її властивості.

У розділі 3 розроблено програмний засіб для реалізації даного алгоритму шифрування, проведено тестування та аналіз його роботи.

Розділ 4 має економічне обґрунтування доцільності розробки.

Пояснювальна записка виконана з дотриманням вимог щодо оформлення магістерських робіт. Ілюстративна частина демонструє функціонування розробленого програмного забезпечення та приклади методу.

Поряд з тим, варто відмітити, в роботі наявні недоліки, а саме, немає порівняльного аналізу ефективності розробленого алгоритму з існуючими аналогами, некоректно описано покроковий алгоритм шифрування, відсутні блок-схеми структури засобу та деталізація окремих етапів тестування.

Не зважаючи на низку зауважень, вважаю, що магістерська кваліфікаційна робота виконана на достатньому рівні і заслуговує на оцінку Е, а її автор – Качай Роман Володимирович заслуговує присвоєння йому кваліфікації: ступінь вищої освіти «магістр», галузь знань – 12 «Інформаційні технології», спеціальність – 125 «Кібербезпека та захист інформації», освітньо-професійна програма – «Безпека інформаційних і комунікаційних систем».

Опонент
к.т.н., доц., доц. каф. ПЗ — (підпис) — Володимир МАЙДАНЮК
(посада, науковий ступінь, вчене звання) (ініціали, прізвище)

ВІДГУК КЕРІВНИКА
на магістерську кваліфікаційну роботу

студента Качая Романа Володимировича
(прізвище, ім'я, по батькові)
на тему Метод та засіб шифрування на основі квазігрупових тотожностей

Сучасніть розвитку інформаційних технологій створює нові ризики для безпеки даних, що є ключовою задачею для забезпечення захисту інформації. Для підтримки належного рівня кіберзахисту є необхідними ефективні методи шифрування, що спрямовані на підвищення стійкості криптографічних алгоритмів. Магістерська кваліфікаційна робота Качая Р.В. присвячена розробці методу та засобу шифрування на основі квазігрупи, яка цікава тим, що відповідний її латинський квадрат незмінний за будь-якої оборотної операції до даної квазігрупи, яка будується за законом Шредера. Ця властивість дає досить низьке навантаження на апаратну частину, цим самим забезпечуючи зменшення апаратних витрат. Крім того, розробка нового підходу до алгоритму шифрування дає можливість підвищити рівень криптографічної стійкості та ефективності захисту інформації.

У межах роботи проведено огляд сучасних підходів до використання квазігруп, латинських квадратів та квазігрупових тотожностней у шифрах. Описано теоретичне обґрунтування нового методу шифрування на основі законів Шредера, які мають записи у вигляді квазігрупової тотожності, що спрощує програмний засіб та оптимізує код програми засобу для шифрування.

Під час виконання магістерської кваліфікаційної роботи студент показав середній рівень теоретичної підготовки, достатнє вміння застосовувати знання на практиці, примітивну здатність самостійно розв'язувати складні технічні завдання та аналізувати отримані результати. Завдання на магістерську кваліфікаційну роботу виконав в повному обсязі. Пояснювальна записка та графічні матеріали підготовлені відповідно до вимог оформлення магістерських робіт. За темою кваліфікаційної роботи студент публікації немає. Проявив себе толерантним та організованим у виконанні поставлених завдань.

Вважаю, що магістерська кваліфікаційна робота виконана на середньому рівні та заслуговує на оцінку «Е», а її автор – Качай Роман Володимирович заслуговує присвоєння йому кваліфікації: ступінь вищої освіти «магістр», галузь знань – 12 «Інформаційні технології», спеціальність – 125 «Кібербезпека та захист інформації», освітньо-професійна програма – «Безпека інформаційних і комунікаційних систем»

Керівник магістерської кваліфікаційної роботи

к.фіз-мат.н., доцент кафедри ЗІ _____ Галина ШЕЛЕПАЛО