ІЛЮСТРАТИВНА ЧАСТИНА

Засіб шифрування даних на основі квазігрупи. Частина 2. Криптографічний алгоритм

Виконав: студент 4 курсу групи 1БС-19б
спеціальності 125 Кібербезпека
_____ Радченко Є. В.
17 червня 2023 р.
Керівник: к.ф.-м.н., доцент каф. ЗІ
_____ Шелепало Г. В.
17 червня 2023 р.

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

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ,
д. т. н., професор
Володимир ЛУЖЕЦЬКИЙ
«28» грудня 2023 року

З А В Д А Н Н Я
НА КОМПЛЕКСНУ БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Радченко Євгенію Валентиновичу

1. Тема роботи: «Засіб шифрування даних на основі квазігрупи. Частина 2. Криптографічний алгоритм»,
керівник роботи Шелепало Галина Василівна, к.ф.-м.н., доцент, затверджено наказом ректора ВНТУ від 20 березня 2023 року №67.
2. Строк подання студентом роботи 17 червня 2023 р.
3. Вихідні дані до роботи:
   - сучасні засоби потокового шифрування;
   - сучасні алгоритми шифрування на основі квазігруп;
   - вимоги до засобів LW-криптографії.
4. Зміст текстової частини: Вступ. 1.Аналіз інформаційних джерел. 2. Метод шифрування на основі CIP-квазігруп. 3. Розробка апаратного засобу для потокового шифрування. Висновки. Список використаних джерел. Додатки.
5. Перелік ілюстративного матеріалу: таблиці Келі парастрофних квазігрупових операцій, таблиця Келі групи Кляйна, список середніх CIP-квазігруп (латинських квадратів), схема алгоритму шифрування, структура засобу шифрування, таблиця з оцінками складності апаратної реалізації засобу.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Шелепало Г. В., к. фіз.-мат. н., доцент	20.03.2023	16.06.2023
2	Шелепало Г. В., к. фіз.-мат. н., доцент	20.03.2023	16.06.2023
3	Шелепало Г. В., к. фіз.-мат. н., доцент	20.03.2023	16.06.2023

7. Дата видачі завдання: 20 березня 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Приміт
1	Аналіз завдання. Вступ	20.03.23–24.03.23	
2	Аналіз інформаційних джерел за напрямком комплексної бакалаврської дипломної роботи	27.03.23–07.04.23	
3	Розробка моделей та алгоритмів	10.04.23–21.04.23	
4	Практична реалізація, моделювання, експериментування, результати	24.04.23–19.05.23	
5	Висновки	22.05.23–24.05.23	
6	Оформлення пояснювальної записки	25.05.23–31.05.23	
7	Попередній захист БКР	01.06.23–08.06.23	
8	Виправлення зауважень, підготовка ілюстративного матеріалу	09.06.23–15.06.23	
9	Представлення БКР до захисту, рецензування	16.06.23–19.06.23	
10	Захист БДР	20.06.23–23.06.23	

Студент _____ Євгеній РАДЧЕ

Керівник роботи _____ Галина ШЕЛЕП

АНОТАЦІЯ

Комплексна бакалаврська дипломна робота складається з двох частин. Дана робота, присвячена першій частині, нараховує 65 сторінок формату А4, на яких є 20 рисунків, 4 таблиці, список використаних джерел, що налічує 40 найменувань.

Бакалаврська робота присвячена розробці апаратного засобу потокового шифрування. В результаті аналізу існуючих способів захисту обрано такий криптопримітив, як середні CIP-квазігрупи 4-го порядку серед ізотопів групи Кляйна з функцією оборотності x^2. Цей криптографічний примітив дозволяє зменшити апаратну складність засобу за рахунок використання ключа меншої розрядності. Після розробки алгоритму засобу потокового шифрування та вибору його модулів здійснено оцінку складності апаратної реалізації та порівняльну оцінку з сучасними апаратними засобами потокового шифрування в LW-криптографії.

Ключові слова: захист інформації, алгоритм шифрування, апаратний засіб шифрування, квазігрупи, парастрофні перетворення, LW-криптографія.

ABSTRACT

The complex bachelor thesis consists of two parts. This work, devoted to the first part, consists of 65 A4 pages format, which have 20 figures, 4 tables, the list of sources used contains 40 items.

Baccalaureate work is dedicated to the development of the hardware problem of stream encryption. As a result of the analysis of the main methods of defense, such a crypto-primitive was selected as the middle CIP-quasigroups of the 4th order of the isotopes of the Klein group with the function of turnover x^2. This cryptographic primitive allows you to change the hardware folding of the account for a small number of keys. After the development of the stream encryption algorithm and the choice of its modules, an assessment of the complexity of the hardware implementation and a similar assessment with the current hardware methods of stream encryption in LW-cryptography were obtained.

Keywords: information protection, encryption algorithm, hardware encryption, quasigroups, parastrophic transformation, LW-cryptography.

ЗМІСТ

ВСТУП

Протягом останнього сторіччя науковий прогрес в усіх сферах здійснив безпрецедентний прорив, надаючи суспільству численні можливості. Особливо важливими для сучасного світу стали останні кілька десятиліть, які визначили теперішній стан людства. Цей період відзначився появою комп'ютерів, мобільних пристроїв, різноманітних ґаджетів та розвитку Інтернету. Ці винаходи покращили якість життя людей, змінили їх спосіб життя та ведення традиційного бізнесу.

В результаті цього науково-технологічного прориву інформація стала найціннішим ресурсом, перевершуючи у своїй вартості золото, алмази, нафту та інші природні ресурси. Інформація стає найбільш важливим товаром і найпотужнішою зброєю. З огляду на зростаючу цінність інформації, очевидним стає факт, що існують люди або групи людей, які прагнуть володіти інформацією, яка не належить їм. Це породжує потребу в захисті інформації.

Оскільки на даний момент існує тенденція до мініатюризації апаратного забезпечення, разом з нею формується необхідність в захисті інформації, що циркулює в малоресурсних системах, таких як мобільні телефони, гарнітури та навіть імпланти та протези. Існує велика кількість методів та апаратних засобів шифрування, створених спеціально для роботи в системах з обмеженими ресурсами. Саме такі засоби шифрування інформації використовує особлива область криптографії – LW-криптографія (від англ. Lightweight - легковаговий). Основною метою LW-криптографії є створення криптографічних алгоритмів, методів, засобів та протоколів, що мають низьку апаратну складність для їх використання в мініатюрних пристроях, де важливим є розмір кожного елемента.

Алгоритми потокового шифрування зосереджуються на необхідності зашифрування невеликих за розміром повідомлень. У випадку зіставлення з блоковими алгоритмами шифрування помітна різниця в ефективності обробки невеликих фрагментів інформації. Також використання потокових шифрів виправдане необхідністю застосування алгоритмів шифрування у випадку, коли

передача конфіденційної інформації призводить до втрати або спотворення цієї інформації. На відміну від блокових шифрів, потокові в такому випадку спотворюють всього один символ, а не цілий блок.

Різними інструментами для потокового шифрування доведена практична цінність та якісне застосування і використання шифрів. Таким математичним апаратом виявилися різноманітні алгебричні структури, зокрема квазігрупи та їх комбінаторні аналоги – латинські квадрати.

Актуальність бакалаврської дипломної роботи обґрунтовує розвиток технологій в напрямку зменшення розмірів апаратури, який створює попит на пристрої, що можуть бути реалізовані в малоресурсних системах. Безпосередньо LW-криптографія охоплює криптографічні алгоритми, що здатні працювати в малоресурсних системах, тобто мають низькі апаратні витрати.

Метою бакалаврської роботи є зменшення апаратних витрат засобу потокового шифрування шляхом розробки методу шифрування на основі середніх CIP-квазігруп 4-го порядку з функцією оборотності x^2 серед ізотопів групи Кляйна. Для досягнення поставленої мети, сформовано такі задачі:

- Проаналізувати сучасні методи потокового шифрування для LW-криптографії.
- Розробити метод шифрування на основі CIP-квазігрупи 4-го порядку серед ізотопів групи Кляйна.
- Розробити структуру засобу для шифрування та оцінити складність його апаратної реалізації.

Об’єктом роботи є процес потокового шифрування. Предметом роботи є метод, алгоритм та засіб для потокового шифрування на основі квазігруп.

Проміжні результати дослідження за темою знайшли своє відображення в роботах, що вказані в Додатку А. Апробацію результатів дослідження було здійснено на наукових конференціях, що вказані в Додатку Б.

1 АНАЛІЗ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

У першому розділі зроблено огляд дослідження галузі малоресурсної криптографії, обґрунтовано актуальність дослідження, описано детальний аналіз проблеми захисту конфіденційності інформації, обґрунтовано необхідність розробки нових алгоритмів шифрування, зроблено аналіз сучасних засобів потокового шифрування для малоресурсних систем, продемонстровано ефективність застосування квазігруп для потокового шифрування.

1.1 Обґрунтування актуальності дослідження

У цифрову епоху, коли обмін інформацією став неухильною складовою нашого повсякденного життя, захист конфіденційності інформації стає все більш важливим та складним завданням. Використання алгоритмів шифрування є одним із найефективніших способів захисту даних та персональної інформації. Проте зростання обчислювальних можливостей та поява нових атак демонструють необхідність постійного розвитку та створення нових алгоритмів шифрування для забезпечення гарантованої конфіденційності інформації.

Перш за все, зростання обчислювальної потужності комп'ютерів вимагає постійного вдосконалення шифрувальних алгоритмів. Старі алгоритми, які раніше вважалися безпечними, можуть бути легко зламані шляхом перебору всіх можливих ключів за короткий період часу, або за допомогою комплексних атак, складених на складних математичних операціях [1]. Створення нових алгоритмів шифрування, які використовують сучасні математичні методи та складні алгоритмічні підходи, дозволяє забезпечити вищий рівень стійкості та безпеки.

Крім того, змінні потреби інформаційного суспільства вимагають створення нових алгоритмів шифрування, які можуть працювати ефективно в різних сценаріях використання. Наприклад, зростання обсягів передачі даних в мережах Інтернету речей та розширення областей застосування блокчейн-технологій ставлять перед науковцями виклик створення шифрувальних алгоритмів, які можуть працювати в

ресурсно-обмежених умовах та забезпечувати конфіденційність даних у розподілених системах [2].

Це проблема, що вимагає поєднання сучасних математичних методів, алгоритмічних підходів та врахування викликів та загроз сучасного цифрового середовища. Тільки шляхом постійного розвитку та створення нових алгоритмів шифрування можна забезпечити надійний та ефективний захист конфіденційності інформації [3], що є фундаментально важливим для сучасного світу.

При проведенні детального аналізу сучасних інформаційних систем можна визначити вразливості та недоліки кожної з цих систем. Використання статистичних даних дозволяє зробити обґрунтований висновок про те, що однією з найважливіших проблем, з якими зіштовхуються малі та середні підприємства в контексті кібербезпеки, є недостатнє використання шифрування як ефективного методу захисту інформації, а також використання слабких або застарілих методів шифрування. Для прикладу, серед найбільших загроз 2021 року за версією дослідження OWASP криптографічні помилки та недоліки посіли друге місце [4].

Відповідно до наявних статистичних даних, можна констатувати, що близько 53% компаній не надають належного рівня захисту важливих файлів, або ж зберігають їх у незашифрованому вигляді [5]. Це свідчить про необхідність удосконалення практик кібербезпеки та просування сучасних криптографічних методів для забезпечення безпеки інформації.

Водночас з цим, провідні технологічні компанії проводять високопрофесійні дослідження того, як неуспішне використання ненадійних алгоритмів шифрування не лише підвищує ризик можливих втрат у разі кібератаки, але також значно ускладнює процес відновлення інформаційної системи в разі несанкціонованого вторгнення. Наприклад, згідно з проведеними експериментами, тривалість відновлення втрачених даних, що виникли внаслідок застосування ненадійних алгоритмів шифрування як методу захисту, складає близько 287 днів [6].

Цей досліджуваний феномен свідчить про необхідність використання сучасних і надійних алгоритмів шифрування в інформаційній безпеці. Такі

алгоритми забезпечують надійний захист конфіденційної інформації та запобігають несанкціонованому доступу до неї. Враховуючи постійні загрози кібербезпеки, дослідники активно працюють над розробкою нових алгоритмів шифрування, які були б неуразливими до атак і забезпечували швидке відновлення системи в разі інциденту. Завдяки цим науковим дослідженням, компанії можуть виявити потенційні слабкі місця і розробити ефективні заходи безпеки для запобігання кібератакам та зменшення часу відновлення після інцидентів. Використання надійних алгоритмів шифрування є критичним елементом в забезпеченні цілісності та конфіденційності інформаційних систем, а також у вдосконаленні безпеки у цифровому середовищі.

Очевидно, що в той час як алгоритми шифрування невпинно розвиваються, так само продовжують розвиватися і методи їх зламу або обходу. Наведена статистика переконливо підтверджує факт необхідності постійних вдосконалень вже існуючих алгоритмів або розробки нових, які будуть відповідати сучасним вимогам в сфері кібербезпеки.

При цьому, з огляду на постійне зростання обчислювальних можливостей, які доступні потенційним зловмисникам, важливо забезпечити високу стійкість алгоритмів шифрування. Для цього необхідно враховувати сучасні методи зламу, такі як крипто-аналітичні атаки, квантові обчислення та атаки на сторонні канали.

Окрім того, зростаюча кількість пристроїв, що підключаються до Інтернету речей, також вимагає розробки нових алгоритмів шифрування, зокрема для забезпечення конфіденційності, цілісності та автентичності даних у мережах.

Проте головна проблема, що виникає при розробці алгоритмів шифрування, полягає в невизначеності наявності ідеального алгоритму потокового шифрування. Хоча існує алгоритм шифрування, що вважається «ідеальним» - One-Time Pad, або ж шифр Вернама, він є абсолютно непрактичним для використання з будь-яким великим фрагментом інформації через необхідність використання та передачі абсолютно випадкового ключа тієї ж довжини, що і відкритий текст [7]. Отже, зберігається потреба в алгоритмах шифрування, що будуть наближеними до

характеристики ідеального і практичними у використанні, надаючи можливість їх реального застосування у малоресурсних пристроях.

1.2 Аналіз галузі малоресурсної криптографії

Технологічний прогрес за останні десятиліття відкрив перед людством незрівнянні можливості у зменшенні розміру обчислювальних засобів. Починаючи з першої електронної обчислювальної машини, відомої як ЕНІАК, що складалася з неймовірно вражаючої кількості 18 тисяч електронних ламп та займав площу декількох кімнат [8], дослідження просунулися настільки далеко, що сучасні процесори мають розмір транзистора всього в 5Нм [9]. Цей феноменальний прогрес не лише дозволив створювати надзвичайно компактні пристрої, але також породив необхідність в кардинальних змінах у підході до розробки апаратного забезпечення.

Для прикладу можна зазначити, що технологія використання електронних ламп майже повністю залишилась в минулому. Винятком стає лише їх використання в якості основи для створення звукової апаратури, де вони мають безсумнівну перевагу за рахунок використання аналогового сигналу без його перетворення в цифровий [10]. В решті сучасного апаратного забезпечення здебільшого використовуються напівпровідникові прилади [11].

Іншим прикладом різких змін в підході до розробки апаратного забезпечення стало введення поняття малоресурсної системи. Апаратним засобом малоресурсної системи називають такий апаратний засіб, чия апаратна реалізація не вимагає великих затрат. Очевидним стає той факт, що в будь-якому пристрої, незалежно від його розміру, необхідно мати засоби захисту інформації. Отже, існує негайна необхідність в апаратних засобах малоресурсної криптографії.

LW-криптографія, або, що теж саме, малоресурсна криптографія, означає розробку та реалізацію криптографічних алгоритмів, які спеціально створені для ефективної роботи та економного використання ресурсів на обмежених пристроях, таких як малопотужні мікроконтролери, вбудовані системи та пристрої Інтернету

речей (IoT), мобільні телефони тощо. Ці пристрої зазвичай мають обмежену обчислювальну потужність, пам’ять та енергетичні ресурси, що й визначає для них спеціальне необхідне шифрування, яке не споживає ні великих технічних потужностей, ні великих об’ємів пам’яті, ні великих енергетичних затрат.

Основна мета полегшеної криптографії — забезпечити надійні гарантії безпеки, мінімізуючи вплив на системні ресурси. Це досягається за допомогою різних методів, включаючи спрощення алгоритму, зменшення розмірів ключів, оптимізоване впровадження та компроміс між безпекою та продуктивністю.

Ось деякі ключові характеристики та обґрунтування легкої криптографії:

– Ефективність: полегшені криптографічні алгоритми створені для ефективних обчислень, вимагаючи менше ресурсів, таких як обчислювальна потужність і пам’ять. Вони спрямовані на досягнення балансу між безпекою та продуктивністю.

– Низьке енергоспоживання: спрощені алгоритми оптимізовано для мінімізації споживання енергії, оскільки багато обмежених пристроїв живляться від батареї та мають обмежений бюджет енергії. Ця оптимізація гарантує, що криптографічні операції можуть виконуватися з мінімальним впливом на загальне енергоспоживання пристрою.

– Невеликий код і об’єм пам’яті: легкі алгоритми розроблені з урахуванням малих розмірів коду та вимог до пам’яті. Це важливо для пристроїв з обмеженою ємністю пам’яті та обмеженими ресурсами пам’яті.

– Стійкість до атак: незважаючи на свою малоресурсну природу, легкі криптографічні алгоритми все одно повинні забезпечувати достатній захист від різних атак, включаючи атаки грубої сили, атаки по бічних каналах і диференціальний аналіз потужності.

– Гнучкість: спрощена криптографія має на меті запропонувати гнучкість щодо вибору алгоритмів, дозволяючи розробникам вибирати відповідні алгоритми на основі їхніх конкретних вимог, включаючи рівень безпеки, продуктивність і обмеження ресурсів [12].

Загалом легка криптографія забезпечує безпечний зв’язок, захист даних і автентифікацію на пристроях з обмеженими ресурсами без суттєвого впливу на їх продуктивність і енергоспоживання. Він відіграє вирішальну роль у забезпеченні швидко зростаючої системи пристроїв Інтернету речей та інших систем.

1.3 Аналіз сучасних засобів потокового шифрування в LW-криптографії

При огляді алгоритмів потокового шифрування, створених безпосередньо для апаратної реалізації, необхідно враховувати результати конкурсу eSTREAM, який був створений для визначення найкращого потокового шифру, який зможе замінити вже застарілі алгоритми шифрування, вразливість яких було доведено теоретично та практично [13].

До переможців конкурсу eSTREAM в категорії апаратних засобів шифрування в першу чергу відноситься родина шифрів Grain. Метод шифрування Grain полягає в використанні двох регістрів зсуву розміром в 80 бітів кожен. Один регістр зсуву має лінійний зворотній зв’язок, а інший – нелінійний зворотний зв’язок. Після цього п’ять значень з регістрів зсуву передаються на вхід бінарної функції $h(x)$ [14]. Перевагою цього алгоритму є простота реалізації з точки зору структури – журі конкурсу eSTREAM назвали родину Grain «елегантною та простою». Хоча в оригінальному шифрі Grain було знайдено вразливість, нова версія Grain v1 отримала не лише збільшену криптостійкість а й варіант алгоритму на основі 128-бітного ключа, на основі якого було створено новий метод під назвою Grain 128a. Також родина шифрів Grain має можливість збільшувати швидкість роботи за рахунок надбудови додаткових регістрів зсуву за рахунок збільшення складності апаратної реалізації. Недоліками цього алгоритму шифрування є необхідність використання 160 тригерів для побудови двох регістрів зсуву та, відповідно, необхідність 160 тактів ініціалізації перед початком безпосередньої роботи засобу [15]. Хоча засіб шифрування, маючи апаратну складність в 1294 GE, підходить під визначення алгоритму шифрування малоресурсної криптографії,

також є місце для вдосконалень а напрямку зменшення витрат його апаратної реалізації.

Іншим переможцем конкурсу eSTREAM став засіб потокового шифрування під назвою MICKEY 2.0. Цей алгоритм шифрування використовує 80-бітний ключ та вектор ініціалізації розміром до 80 бітів. Суть роботи засобу полягає в використанні двох 100-бітних регістрів зсуву, один з яких є лінійним, а інший – нелінійним, кожен з яких несинхронно виконує операцію зсуву під впливом іншого [16]. До переваг подібного методу шифрування можна привести використання асинхронних регістрів зсуву, що не дозволяє використати звичайні методи криптографічного аналізу, збільшуючи загальну криптостійкість засобу. До недоліків засобу потокового шифрування можна віднести відносно високу складність апаратної реалізації в зв'язку з використанням двох 100-бітних регістрів зсуву. І хоча алгоритм, маючи апаратну складність в 1846 GE, підходить під визначення засобу потокового шифрування малоресурсної криптографії, на даний момент не є доведеною можливість його спрощення через необхідність в регістрах зсуву, що перевищують розмір ключа в 2,5 рази [17].

Останнім переможцем конкурсу eSTREAM в категорії апаратних засобів потокового шифрування став метод шифрування під назвою Trivium. Характерною рисою алгоритму є його висока здатність до паралелізації, що дозволяє водночас обробляти до 64 бітів повідомлення за рахунок відносно невеликого збільшення складності апаратної реалізації. Trivium використовує 80-бітний ключ та 80-бітний вектор ініціалізації. Внутрішній стан засобу задається трьома взаємно поєднаними регістрами зсуву з нелінійним зворотнім зв'язком, які мають розмір в 93, 84 та 111 бітів відповідно. Ініціалізація засобу є доволі схожою за алгоритмом до генерації шифротексту та займає 1152 такти. Згідно заяв авторів засобу, Trivium здатний згенерувати 2^{64} бітів шифротексту з кожної пари ключ/вектор [18]. До переваг засобу потокового шифрування Trivium можна віднести його високу здатність до паралелізації – при збільшенні кількості потоків з 1 до 64 складність апаратної реалізації зростає з 2599 GE до всього 4921 GE [19]. До недоліків можна віднести

апаратну складність, що не дозволяє віднести Trivium до засобів шифрування малоресурсної криптографії. Хоча існували спроби створити аналоги Trivium меншого розміру, наприклад, варіація Bivium, що має два регістри зсуву з нелінійним зворотнім зв'язком замість трьох, або варіанти зі зменшеним розміром регістрів зсуву, всі ці варіанти оригінального засобу були зламані, в той час як використані атаки виявились неефективними проти повної версії апаратного засобу потокового шифрування Trivium.

Іншим алгоритмом потокового шифрування є Enocoro-128. Цей засіб потокового шифрування оснований на двох 32-бітних буферах з регістрами зсуву, п'яти блоках перестановок та операції виключного АБО. Засіб Enocoro-128 здатний шифрувати 1 байт за такт та використовує 128-бітний ключ та 64-бітний вектор ініціалізації. За співвідношенням швидкодії до апаратної складності алгоритм Enocoro-128 є практично рівним алгоритму Trivium та родині шифрів Grain. До недоліків цього засобу потокового шифрування можна віднести високу апаратну складність, що становить 4100 GE (2700 GE для засобу потокового шифрування Enocoro-80, що працює на ключі розміром 80 біт).

Алгоритм потокового шифрування F-FCSR-H був представлений на конкурсі eSTREAM, але пізніше був виключений з eSTREAM Portfolio після того, як в методі була знайдена вразливість. Сутність родини потокових шифрів F-FCSR полягає в використанні регістра зсуву зі зворотнім зв'язком по переносу та лінійного фільтру F-FCSR-H має довжину головного регістра 160 біт, тривалість ініціалізації 182 такти та статичний фільтр. Апаратний засіб потокового шифрування F-FCSR-H не підходить під визначення малоресурного криптографічного засобу через високу апаратну складність, яка становить 4800 GE, та критичну вразливість, що полягає в можливості перетворення нелінійної функції FCSR в лінійну [20].

На даний момент Salsa20 вважається одним з найкращих потокових шифрів. Хоча початковий варіант алгоритму був створений виключно для категорії програмної реалізації конкурсу eSTREAM, також був представлений і варіант алгоритму у вигляді апаратного засобу. Алгоритм працює Здебільшого засіб

потокового шифрування Salsa20 не поступався переможцям конкурсу в категорії апаратних засобів потокового шифрування [21], але його апаратна складність становила 3842 GE, що не дозволяє назвати цей засіб потокового шифрування, що доцільно використовувати для LW-криптографії.

Засіб потокового шифрування LIZARD оснований на роботі двох регістів зсуву з нелінійним зворотнім зв'язком. Один з них має розмір внутрішнього стану 90 бітів, а інший – 31 біт. Ці регістри зсуву є взаємопов'язаними за допомогою операції виключного АБО, що принципово є схожим до алгоритмів родини Grain. Перевагою перед вищевказаною родиною шифрів є менший розмір регістрі зсуву, що дозволило зменшити апаратну складність реалізації даного алгоритму шифрування до 1161 GE [22].

Fruit-80 має будову, схожу до алгоритмів Grain та LIZARD, але додатково використовує додаткову бінарну функцію *g(.)*, що проводить операції додавання за модулем два з ключем та значеннями заздалегідь визначених регістрів з регістру зсуву з нелінійним зв'язком [23]. Розмір регістрів зсуву становить 43 та 37 бітів відповідно, що зменшує загальну апаратну складність цього алгоритму до 960 GE. Проте недоліком є неможливість повторного використання вектора ініціалізації, навіть з іншим секретним ключем.

Алгоритм Plantlet є схожим до алгоритму Fruit-80, але додатково містить функцію *f*, яка впливає на процес ініціалізації регістра зсуву з лінійним зворотнім зв'язку. Також відмінністю є змінений поліном, на основі якого побудовано регістр зсуву з лінійним зворотнім зв'язком та вихідна функція *h*. Загальна складність реалізації цього засобу становить 928 GE, проте було визначено декілька атак на цей засіб потокового шифрування, які використовують відновлення ключа шляхом розв'язання поліноміальних рівнянь та передбаченні стану регістрів зсуву [24].

Засіб потокового шифрування LILLE, згідно заяви авторів, є легковаговим шифром, що оснований на короткому внутрішньому стані. З регістра зсуву на нелінійну функцію передається послідовність. Частина ключа разом з повідомленням застосовується в цій нелінійній функції, інша накладається на

результат її роботи в якості гами для повернення в регістр стану та повторного використання [25]. Перевагою засобу шифрування LILLE є надзвичайна простота апаратної реалізації – майже рекордні 911 GE. Проте дуже швидко було знайдено атаку на алгоритм з необхідним часом $2^{41.2}$.

1.4 Характеристика квазігруп як криптографічного примітиву

Теоретично доведено ефективність застосування квазігруп в якості криптографічних примітивів [26-31]. В підтримку цієї ідеї можна навести такі характеристики квазігруп та їх парастрофних перетворень:

- Потенційно незліченна кількість квазігруп, наприклад починаюи вже з 12 порядку досі не відома їх точна кількість [26].
- Велика кількість можливих оборотних перетворень для кожної квазігрупи. У бінарному випадку, таких перетворень не більше шести [27].
- Оберненість та спряженість операцій над квазігрупами за рахунок різних видів симетричності. Для бінарних квазігруп таких різновидів всього чотири: тотальна симетричність, напівсиметричність, односиметричність, асиметричність. Односиметричність, в свою чергу, має види комутативність, ліва симетрія та права симетрія. Всі три види симетрії є спряженими операціями [28].
- Подання квазігрупових операцій у вигляді латинських квадратів, кубів, гіперкубів, графів, граток, номограм, сіток та інших математичних структур [29]

Ці та інші властивості квазігруп дозволяють використання в багатьох сферах криптографії, зокрема, в якості основи для створення алгоритмів шифрування, побудови кодів, геш-функцій, функцій обміну ключами тощо.

Як визначено в [30], «Квазігрупа – це латинський квадрат, що задовольняє певну рівність, якою вона і визначається. В свою чергу латинський квадрат – це масив $n \times n$, де кожний елемент зустрічається в кожному рядку та стовпчику всього раз. Розмір квазігрупи називається «порядком», наприклад, квазігрупа 16-го порядку – латинський квадрат розміром 16×16.

Особлива властивість квазігруп, що дозволяє побудувати на їх основі алгоритми шифрування – це надзвичайно велика кількість квазігруп, що збільшується експоненціальною властивістю зі збільшенням їх порядку. Так, наприклад, кількість квазігруп 5-го порядку – 1411, шостого – 1,130,531, а от вже одинадцятого можливих

19,464,657,391,668,924,966,791,023,043,937,578,299,025,

тобто близько 10^{40}. Кількість квазігруп порядків більших за 11 досі не обчислена, але спостерігаючи за тенденцією зростання можна припустити, що кількість квазігруп навіть 16-го порядку буде достатньою для використання їх в якості основи для алгоритму шифрування.

В реалізації шифру використовується набір бінарних операцій і псевдовипадкова послідовність їх виконання, яка залежить від секретного ключа, що забезпечує додаткову секретність в процесі шифрування. Ці операції мають бути різними за своїми властивостями і простими в реалізації. Саме такий набір операцій забезпечують бінарні квазігрупи 4-го порядку [31].

Згідно з класифікацією парастрофної симетрії [28] всі бінарні квазігрупи поділяються на:

- тотально симетричні, коли всі парастрофи їх збігаються, тобто в результаті перетворень маємо один латинський квадрат;
- напівсиметричні, коли є два різних парастрофи, тобто в результаті перетворень маємо два різних латинських квадрати;
- Односторонньо-симетричними, коли є три різних парастрофи, тобто в результаті перетворень маємо три різних латинських квадрати;
- Асиметричні, коли є шість різних парастрофів, тобто в результаті перетворень маємо шість різних латинських квадратів.

Односторонньо-симетричних парастрофів є три види:

- комутативні;
- ліво-симетричні;
- право-симетричні.;

Всі вони спряжені між собою, тобто існує ланцюг перетворень від одного парастрофа до іншого.

Оскільки деякі квазігрупи є спряженими одна до одної, в залежності від їх класифікації, тому апаратна реалізація симетричного алгоритму потокового шифрування, основаного на квазігрупах, може бути виконана за рахунок властивостей спряженості та оборотності квазігруп..

В результаті бачимо, що на додаток до великої кількості квазігруп вищих порядків та кількох операцій, що можуть бути застосовані до них, є також чітка класифікація за властивістю симетричності, оборотності та спряженості квазігруп, що може бути використано для створення різних криптографічних примітивів та повноцінних алгоритмів шифрування.

1.5 Приклади застосування квазігруп у шифруванні

Класична схема Ель-Гамаля полягає в використанні криптосистеми з відкритим ключем, принцип якої ґрунтується на складності обчислення дискретних логарифмів. Оригінальна схема працює за методом множення простих чисел за модулем. Існує модифікація схеми Ель-Гамаля, що використовує бінарну квазігрупу 5-го порядку та три її парастрофи для за шифрування.

Можливим застосуванням квазігруп в криптографії є використання їх для побудови алгоритму обміну ключами. Деякі із запропонованих нещодавно алгоритмів називаються MQQ (Multivariate Quadratic Quasigroups) та MCQ (Multivariate Cubic Quasigroups), вони ґрунтуються на багатоваріантності квадратичних або кубічних поліномів і квазігрупових перетворень [32].

Очевидним є те, що така кількість можливих квазігруп унеможливлює прямі атаки на криптографічний алгоритм з їх застосуванням, що ставить алгоритми шифрування на основі квазігруп на один рівень з провідними існуючими алгоритмами шифрування.

Відомими є деякі способи реалізації алгоритмів потокового шифрування на основі квазігруп та алгоритму Ель-Гамаля. Запропонований варіант не лише є

доведенням теоретичної можливості побудови криптографічних алгоритмів, основаних на квазігрупах та їх перетвореннях, а й показує, що подібні алгоритми можуть мати надзвичайну криптографічну стійкість. Проте цей алгоритм, запропонований Данілом Глігороським, залежить від здатності швидко обчислювати великі прості числа, що дещо збільшує затрати часу [33].

Іншим відомим прикладом алгоритму потокового шифрування на основі квазігруп є Edon80, алгоритм, що був розроблений трьома авторами: Данілом Глігороським разом з Смайлом Марковським та Свейном Йоханом Кнапскогом в Норвежському Інституті науки та технологій. Оскільки алгоритм розроблявся спеціально для апаратної реалізації з головною метою зменшення споживання енергії за рахунок зменшення кількості логічних вентилів, він був позитивно сприйнятий конкурсним проектом eSTREAM. Автори запропонували його з метою дослідження та аналізу алгоритмів потокового шифрування [34].

Аалгоритм Edon80 неодноразово піддавався покращенням та модифікаціям. Так, у 2007 році була створена версія алгоритму, здатна працювати з комп'ютерами MAC. Пізніше, була проведена спроба атаки на алгоритм за допомогою розгляду структури його, як алгоритму перестановки або послідовності бінарних операцій. Не дивлячись на те, що були знайдені деякі цікаві залежності, провести повноцінну атаку на алгоритм не вдалося [35].

Ще було проведено дослідження можливості атаки за допомогою відновлення ключа. Як результат виявлено, що частину ключа можна відновити зі складністю 2^{69}. Як висновок, автори ідеї заявили, що алгоритм доволі стійкий до подібного типу атак [36].

Описані факти, в свою чергу, доводять про значну надійність такої математичної моделі, як квазігрупи (латинські квадрати) у вигляді криптографічного примітиву та теоретичну можливість створення ефективних алгоритмів потокового шифрування, основаних на квазігрупах.

Таким чином, спостерігається тенденція щодо ефективного застосування квазігруп в якості основи для побудови алгоритмів шифрування, кодування та

обміну ключами. Це показує можливості використання квазігруп в якості криптографічного примітиву та прикладом побудови криптографічних алгоритмів з різними характеристиками та різним призначеннями.

Отже, було проведено аналіз середовища, визначено проблему, актуальність та попередні розв'язки визначених задач. Наступним було запропоновано альтернативний приклад, проаналізовано теоретичне обґрунтування для нього та наведено практичне застосування. В якості висновку до розділу можна зазначити, що нові підходи до проблеми захисту конфіденційності інформації у вигляді алгоритмів потокового шифрування, основаних на квазігрупах, є актуальними. Ефективність використання квазігруп як криптографічного примітиву було доведено і теоретично, і практично. При цьому, враховуючи тенденції розвитку криптографії, можна зазначити, що потреба в нових алгоритмах шифрування зберігається, що доводить актуальність теми дипломної роботи.

Висновки до розділу

На основі проаналізованих джерел інформації та огляду предметної області було сформовано висновки.

Можна з впевненістю сказати, що галузь малоресурсної криптографії є надзвичайно важливою в сучасному світі за рахунок стрімкого розвитку технологій та тенденції до мініатюризації пристроїв.

Так, у випадку, якщо найближчим часом будуть розроблені робочі прототипи нейронних імплантів, з'явиться гостра необхідність в засобах захисту інформації, яка обробляється та зберігається в цих імплантах.

Таким чином, вже зараз з'являється необхідність в розробці засобів захисту інформації в малоресурсних середовищах. Область криптографії під назвою малоресурсна криптографія або LW-криптографія займається створенням засобів, методів та алгоритмів захисту інформації, таких як протоколи автентифікації, геш-функції, алгоритми шифрування, які будуть мати мінімальний плив на енергоресурси апаратного засобу, його швидкодію та розмір.

При аналізі сучасних апаратних засобів потокового шифрування було визначено, що більшість з них використовують фіксовану послідовність виконуваних операцій, що призводить до необхідності використання секретних ключів збільшеної розрядності, що в свою чергу призводить до збільшення апаратної складності засобів.

На доданок, більшість з апаратних засобів потокового шифрування мають апаратну складність більше ніж 2000 GE, що не дозволяє класифікувати їх як апаратний засіб малоресурсної криптографії. Решта ж методів мають складність реалізації близько 1000-1500 GE, що підходить під параметри LW-криптографії, проте не є ідеальним.

Після аналізу існуючих алгоритмів шифрування на основі квазігруп було визначено, що квазігрупи, як криптографічний примітив, є перспективними, що було доведено теоретично та навіть практично – деякі алгоритми шифрування на основі квазігруп стали учасниками та навіть фіналістами конкурсу eSTREAM, як, наприклад, апаратний засіб потокового шифрування Edon80.

Поданий огляд літературних джерел та аналіз алгоритмів шифрування демонструє чітке обґрунтування та актуальність використання бінарних квазігруп 4-го порядку зі спеціальними властивостями оборотності та спряженості для реалізації нового алгоритму потокового алгоритму шифрування.

2 МЕТОД ШИФРУВАННЯ НА ОСНОВІ CIP-КВАЗІГРУП

Другий розділ присвячений розробці методу та створення алгоритму шифрування на основі середніх CIP-квазігруп 4-го порядку з функцією оборотності x^2 серед ізотопів групи Кляйна.

2.1 Опис криптографічного примітиву на основі квазігрупи

Квазігрупою називається групоїд $(Q;\cdot)$ такий, що для довільних a, b ϵ Q система рівнянь:

$$a \cdot x = b,$$

$$y \cdot a = b$$

має єдиний розв'язок. Квазігрупа $(Q;\cdot)$ називається: середньою, лівою та правою CIP-квазігрупою (CIP – cross inverse properties, в перекладі означає «властивість схрещеної оборотності»), якщо відповідно існують відображення *h, t, b* такі, що для всіх *x, y* виконуються рівності:

$$h(x) \cdot xy = y,$$

$$yx \cdot y = t(x),$$

$$y \cdot xy = b(x),$$

де h, t, b називаються лівою, правою та середньою функцією оборотності.

В якості основи для побудови алгоритму потокового шифрування були використані результати перетворень, описані над ізотопами групи Кляйна для середніх CIP-квазігруп 4-го порядку з функцією оборотності x^2, що задовольняють таких тотожностям [37]:

$$x \cdot yx^2 = y,$$

$$xy \cdot x^2 = y,$$

$$x^2y \cdot x = y,$$

$$x^2 \cdot yx = y.$$

Нагадаємо, що групою Кляйна називається множина з визначеною операцією $\oplus$, яка означає додавання за модулем і виконується рівність

$$\bar{x} \oplus \bar{x} = \overline{0},$$

де $\bar{x}$ означає вектор, координатами якого є елементи множини $\{0;1\}$.

Кожна бінарна квазігрупова операція має парастрофні перетворення, які називаються парастрофами головної квазігрупової операції. Їх в бінарному випадку всього є шість :

– головна операція;
– ліве ділення;
– праве ділення;
– комутування;
– комутування правого ділення;
– комутування лівого ділення;

Нехай в бінарному випадку, головна операція зображена рівністю:

$$x \cdot y = z, \tag{2.1}$$

де під x розуміються елементи рядка таблиці Келі відношення елементів квазігрупи, а під y – елементи стовпця, а під результатом z зазначеної рівності – елемент на перетині рядка x та стовпця y.

Парастроф лівого ділення квазігрупової операції полягає у звичайній перестановці змінних операції у рівності (2.1) яка є головним парастрофом квазігрупової операції. Змінні x та z міняються місцями, в результаті чого отримується рівність:

$$z \cdot y = x$$

Парастроф правого ділення є схожою до парастрофа лівого ділення, проте для перестановки беруться змінні *y* та *z*. В результаті отримуємо рівність:

$$x \cdot z = y$$

Відповідно за цими рівностями можна побудувати нову квазігрупу.

Парастроф комутування полягає в перестановці місцями змінних *x* та *y*. В результаті отримуємо рівність $y \cdot x = z$. Оскільки квазігрупи, на відміну від груп, не є асоціативними, така перестановка дозволяє отримувати абсолютно нову квазігрупу, відміну від головного парастрофа квазігрупової операції.

Парастрофи комутування правого ділення та комутування лівого ділення полягають у використанні пара строфа комутування окремо до лівого ділення або до правого ділення відповідно. Аналогічно, в результаті відповідної перестановки отримуємо рівності:

$$y \cdot z = x,$$

$$z \cdot x = y,$$

що відповідають решті парастрофів бінарної квазігрупової операції. Кількість різних парастрофів дозволяє будувати нові квазігрупи одної й тої ж самої головної квазігрупової операції.

Таким чином, кожна бінарна квазігрупа може мати 6 парастрофних перетворень. В залежності від використання, для побудови алгоритму шифрування можу бути використана будь-яка з них.

У випадку даного алгоритму потокового шифрування для зашифрування було обрано парастроф головної операції, а для розшифрування – парастроф лівого ділення даної головної квазігрупи.

Процес зашифрування можна представити у вигляді рівності головної операції квазігрупи, де змінна *x* заміняється на фрагмент повідомлення M_i, а змінна *y* заміняється на фрагмент гами G_i. Їх результат *z* перетину рядка і стовпця латинського квадрата замінюється фрагментом шифротексту C_i. Як наслідок, отримуємо рівність:

$$M_i \cdot G_i = C_i, \qquad (2.2)$$

де операція (·) визначає головний парастроф визначеної квазігрупової операції.

Для процесу розшифрування використовується операція лівого ділення бінарної квазігрупової операції. При цьому в рівності (2.1) яка визначає головний парастроф квазігрупової операції, змінні *x* та *z* міняються місцями. Відповідно, у випадку використовуваної рівності для шифрування (2.2) в результаті виконання операції лівого ділення отримуємо рівність:

$$C_i \cdot G_i = M_i,$$

де операція (·) вже визначає парастроф лівого ділення даної квазігрупової операції.

Застосування однієї й тієї ж самої квазігрупової операції дозволяє використовувати одні побудовані її парастрофи, подані у вигляді латинських квадратів для зашифрування, а інші її парастрофи – для розшифрування. Це, в свою чергу, спричинює значні зменшення складності реалізації алгоритму шифрування інформації.

2.2 Ідея алгоритму шифрування та його властивостей

Вхідними даними для алгоритму шифрування є повідомлення M та секретний ключ K:

$$M=\{m_1, m_2, m_3, \ldots, m_n\},$$

де *n* означає кількість блоків в повідомленні, кожний з яких має довжину 2.

Кожен символ повідомлення M буде розбитий на пари бітів, що сформують блоки, до яких будуть застосовуватись криптографічні операції.

Секретний ключ K має дві складові:

$$K=\{k_1, k_2\},$$

Перша складова k_1 використовується в якості зерна для генерування послідовності:

$$P=\{p_1, p_2, p_3, \ldots, p_n\}$$

на основі регістру зсуву.

Регістр зсуву - це електронний пристрій, що використовується в цифрових системах для зберігання та передачі даних у вигляді послідовності бітів. Він складається з послідовно з'єднаних тригерів (звичайно, D-тригерів), які здатні зберігати один біт інформації. При подачі керуючого сигналу тригери передають своє поточне значення в наступний тригер. Цей процес називається зсувом.

Друга складова k_2 використовується в якості зерна для генерування псевдовипадкової послідовності:

$$J=\{j_1, j_2, j_3, \ldots, j_n\},$$

де j_i визначає квазігрупу з набору:

$$L = \{L_0, L_1, L_3, L_5\},$$

ту, що буде використана для кожної окремої операції всередині алгоритму. Індекси квазігруп означають відповідні латинські квадрати із таблиць Келі відповідних квазігрупових операцій, зображених в таблиці 2.1

Таблиця 2.1 – Латинські квадрати групи Келі

Основний набір	**0)**	0	1	2	3	**1)**	0	1	2	3	**3)**	0	1	2	3	**5)**	0	1	2	3
	0	0	3	1	2	0	1	2	0	3	0	2	1	3	0	0	3	0	2	1
	1	2	1	3	0	1	3	0	2	1	1	0	3	1	2	1	1	2	0	3
	2	3	0	2	1	2	2	1	3	0	2	1	2	0	3	2	0	3	1	2
	3	1	2	0	3	3	0	3	1	2	3	3	0	2	1	3	2	1	3	0
Побічний набір	**7)**	0	1	2	3	**2)**	0	1	2	3	**4)**	0	1	2	3	**6)**	0	1	2	3
	0	0	2	3	1	0	1	3	2	0	0	2	0	1	3	0	3	1	0	2
	1	3	1	0	2	1	2	0	1	3	1	1	3	2	0	1	0	2	3	1
	2	1	3	2	0	2	0	2	3	1	2	3	1	0	2	2	2	0	1	3
	3	2	0	1	3	3	3	1	0	2	3	0	2	3	1	3	1	3	2	0

Результатом зашифрування є послідовність:

$$C=\{c_1, c_2, c_3, \ldots, c_n\},$$

де:

$$c_i = L_{j_i}(m_i, p_i),$$

є результатом перехрестя рядка m_i та стовпчика g_i з таблиці Келі відповідної квазігрупи L_{j_i}.

Процес розшифрування повідомлення здійснюється за оборотною квазігрупою з множини відповідних квазігруп:

$$L^{-1} = \{L_2, L_4, L_6, L_7\},$$

де кожний елемент повідомлення обчислюється за формулою:

$$m_i = L_{j_i}^{-1}(c_i, p_i),$$

Оборотність алгоритму основується на використанні взаємно-обернених квазігруп для зашифрування та розшифрування.

Оскільки головною метою будь-якого алгоритму шифрування є забезпечення конфіденційності інформації та недопущення зламу алгоритму зловмисником, основною вимогою до алгоритму шифрування на основі CIP-квазігруп 4-го порядку буде висока криптостійкість. Додатковими вимогами буде швидкодія та зниження складності реалізації алгоритму.

Криптостійкість алгоритму полягає не тільки в псевдовипадковості параметрів криптографічних операцій, а й у псевдовипадковості самих операцій. Оскільки конкретна квазігрупа, що буде використана, залежить від псевдовипадкової послідовності *J*, а стовпчик, який буде використано залежить від гами *G*, криптостійкість алгоритму значно збільшується.

2.3 Розробка методу потокового шифрування

Для розробки алгоритму шифрування необхідно створити словесний алгоритм, що описує процес роботи апаратного засобу потокового шифрування. Для побудови словесного алгоритму розглянемо два режими роботи: зашифрування та розшифрування.

Загальний алгоритм роботи описаний нижче.

Першим кроком у процесі реалізації даного алгоритму є введення ключів в регістр зсуву та генератор псевдовипадкової послідовності. Цей етап включає ініціалізацію засобу шляхом введення ключів паралельно відносно один одного, але послідовно у випадку розгляду конкретного модуля апаратного засобу потокового шифрування.

Додатково, слід зазначити, що процес ініціалізації є обов'язковим, і не може бути пропущений з огляду на необхідність заповнення всіх регістрів зсуву апаратного засобу шифрування вхідними даними, в даному випадку – частинами секретного ключа.

Другим кроком буде визначення режиму роботи засобу за рахунок керуючого сигналу, що надходить на блок керування. Отримуючи сигнал «зашифрування» блок керування передає відповідні сигнали на операційний блок, що дозволяє використання квазігруп з набору L. У випадку отримання керуючого сигналу «розшифрування» блок керування передає відповідний сигнал на вхід операційного блоку, що визначає поточну операцію як розшифрування, використовуючи квазігрупи з набору L^{-1}.

Далі відбувається генерування псевдовипадкового числа з псевдовипадкової послідовності. Генерування ПВП здійснюється генератором псевдовипадкової послідовності на основі регістра зсуву. Для кожної окремої операції генератор передає на вихід два біти псевдовипадкової послідовності. Далі ця пара бітів надходить на блок керування, який, на основі отриманих даних, визначає конкретну

квазігрупу з набору L або L^{-1}, в залежності від керуючого сигналу «зашифрування/розшифрування».

Наступним кроком в операційний блок надходять змінні M_i та K_i. Вони надходять у вигляді пар бітів з входу апаратного засобу та регістру зсуву відповідно.

П’ятим кроком є виконання операції зашифрування або розшифрування, в залежності від стану контролюючого сигналу. Операційний блок виконує одну з чотирьох операцій, що визначені заздалегідь. На виході операційного блоку отримуємо фрагмент шифротексту C_i, що складається з двох бітів, або фрагмент повідомлення M_i, в залежності від режиму роботи апаратного засобу потокового шифрування, що.

Після виконання операції зашифрування або розшифрування регістр зсуву нециклічно зсуває ключ K на два біти. На місце зсунутих бітів передаються біти відкритого повідомлення M_i. Визначення бітів, що передаються в регістр зсуву відбувається за рахунок комутатора, що контролюється блоком керування в залежності від керуючого сигналу «зашифрування/розшифрування».

У випадку, якщо був зашифрований або розшифрований весь обсяг повідомлення, апаратний засіб завершує роботу. У випадку, якщо обсяг повідомлення не було вичерпано, алгоритм повертається на етап генерації псевдовипадкового числа.

2.4 Розробка алгоритму потокового шифрування

Для розробки алгоритму роботи засобу потокового необхідно створити словесний алгоритм роботи. Для початку описано словесний алгоритм процесу защифрування, а потім словесний алгоритм процесу розшифрування.

Словесний алгоритм процесу зашифрування:

1) Введення ключів K_1 та K_2
2) Генерування нового числа з псевдовипадкової послідовності
3) Вибір квазігрупи з набору L
4) Отримання на вхід операційним блоком M_i та P_i

5) Виконання головної операції квазігруп
6) Зсув послідовності P
7) Запис використаних бітів повідомлення в регістр зсуву.
8) Якщо обсяг повідомлення не вичерпано, то повертається до пункту 2

Алгоритм роботи апаратного засобу потокового шифрування в режимі «розшифрування» є практично ідентичним алгоритмом роботи режиму «зашифрування».

Словесний алгоритм розшифрування:

1) Введення ключів K_1 та K_2
2) Генерування нового числа з псевдовипадкової послідовності
3) Вибір квазігрупи з набору L^{-1}
4) Отримання на вхід операційним блоком C_i та P_i
5) Виконання квазігрупової операції лівого ділення
6) Зсув послідовності P
7) Запис отриманих бітів повідомлення в регістр зсуву.
8) Якщо обсяг повідомлення не вичерпано, повертається до пункту 2

Таким чином, можна приступати до розробки алгоритму роботи апаратного засобу потокового шифрування. Схема алгоритму зображена на рисунку 2.1

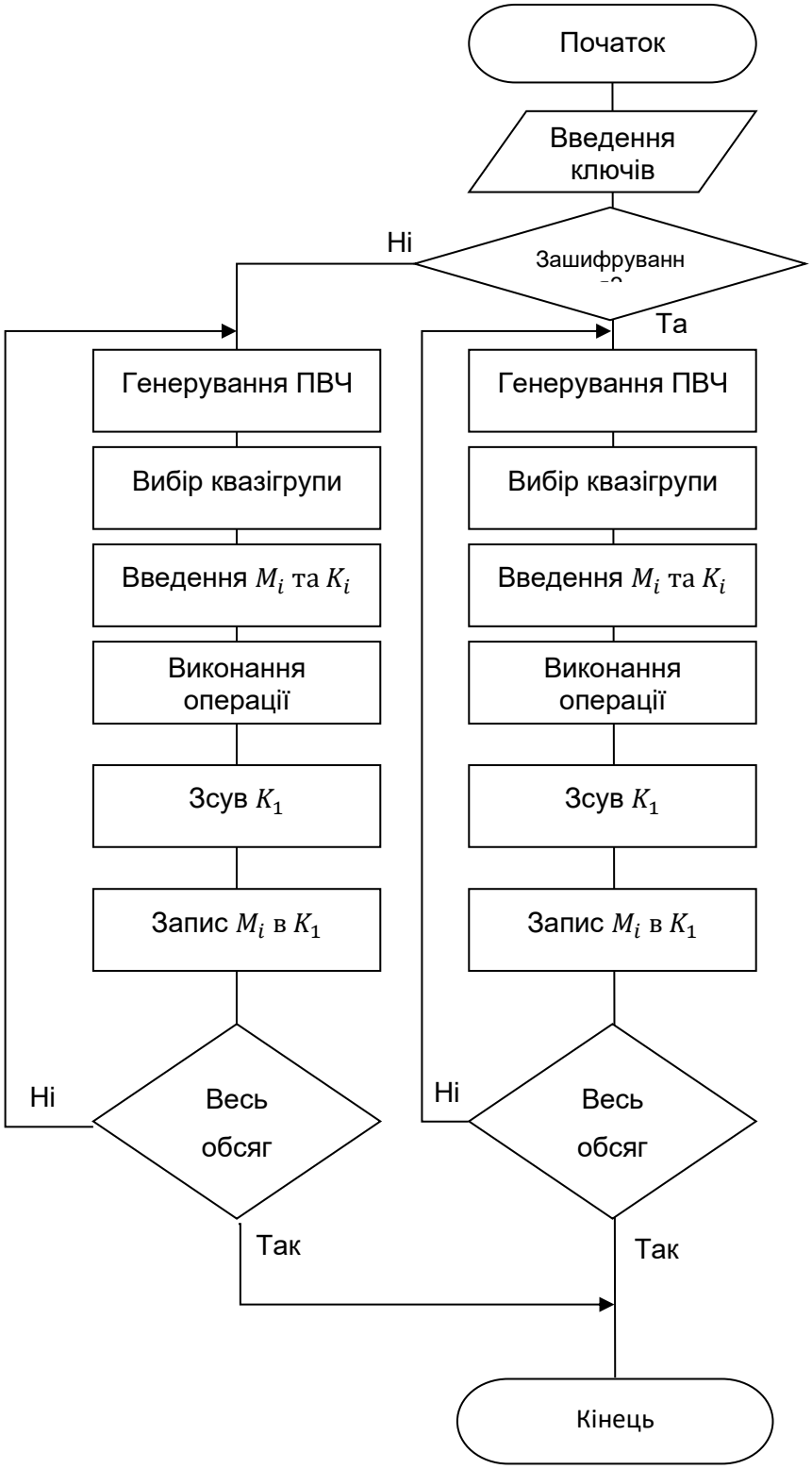


Рисунок 2.1 – Алгоритм засобу потокового шифрування

Генератор псевдовипадкової послідовності передає псевдовипадкове число до блоку керування, регістр зсуву визначає параметр поточної квазігрупової операції, блок керування визначає режим роботи та поточну квазігрупову операцію на основі псевдовипадкового числа, операційний блок обчислює результат квазігрупової операції, обраної блоком керування, на основі вхідних даних та стану регістра зсуву, а комутатор визначає, які дані надходять на регістр зсуву.

Генератор псевдовипадкової послідовності - це алгоритм або пристрій, який створює послідовність чисел або бітів, які, за виглядом, нагадують випадкову послідовність, але фактично є детермінованими, оскільки їх можна відтворити з використанням початкового значення, відомого як "зерно".

Регістр зсуву - це електронний пристрій, що використовується в цифрових системах для зберігання та передачі даних у вигляді послідовності бітів. Він складається з послідовно з'єднаних тригерів (звичайно, D-тригерів), які здатні зберігати один біт інформації.

Операційний блок виконує квазігрупові операції за рахунок реалізованих логічних виразів, що їх описують. Окрім схем, що реалізують квазігрупи, операційний блок також містить комбінаційну схему, яка керує правильністю виводу результату роботи операційного блоку.

Комутатор визначає, які дані будуть надходити до регістру зсуву. Це, в залежності від режиму роботи, можуть бути біти, що надходять на вхід засобу потокового шифрування, або результат роботи операційного блоку

2.5 Розгляд прикладу роботи операційного блоку

Розглядається алгоритм роботи операційного блоку алгоритму шифрування. Для прикладу візьмемо повідомлення $M=\{10011100\}$. Оскільки операційний блок приймає на вхід 2 біти вхідного повідомлення, що шифруються окремо, вхідне повідомлення M розбивається на частини M_1, M_2, M_3 та M_4 які дорівнюють, відповідно, “10”, “01”, “11” та “00”. Припустимо, що послідовність $P=\{01100011\}$. Аналогічно до повідомлення, гама G має надходити до операційного блоку в вигляді

пари бітів, отже вона ділиться на частини P_1, P_2, P_3, P_4, де P_1 = “01”, P_2 = “10”, P_3=”00”, P_4=”11”. Псевдовипадкова послідовність, яка надходить на блок керування, визначає квазігрупу, що буде використана для поточної операції. Оскільки як для зашифрування так і для розшифрування використовуються 4 квазігрупи, то псевдовипадкова послідовність також ділиться на пари бітів. Таким чином, припустимо, що J={01111000}, а після поділу J_1 = ”01”, J_2 = “11”, J_3=”10”, J_4=”00”. Це означає, що для першої операції використовується квазігрупа L_1, для другої – L_3, для третьої – L_2, а для четвертої – L_0.

Результат роботи алгоритму шифрування – шифротекст, що буде позначений C. Оскільки розгляд прикладу проводиться на чотирьох операціях зашифрування та чотирьох операціях розшифрування, матимемо шифротекст C= {C_1, C_2, C_3, C_4}. Квазігрупа L_1, що використовується для першої операції, наведена на рисунку 2.1:

L_1	00	01	10	11
00	00	11	01	10
01	10	01	11	00
10	11	00	10	01
11	01	10	00	11

Рисунок 2.1 – Таблиця Келі Квазігрупи L_1

Процес зашифрування полягає в застосуванні основної квазігрупової операції $x \cdot y = z$. В роботі алгоритму змінна x заміняється на M_i, змінна y заміняється на P_i, а змінна z – на C_i. В результаті маємо формулу:

$$M_i \cdot P_i = C_i, \tag{2.3}$$

де M_i є номером рядка, P_i є номером стовпчика, а C_i є результатом на перетині цих рядка та стовпчика.

Підставляючи в формулу (2.3) значення M_1=”10” та P_1,=”01” отримуємо значення C_1= “00”, як показано на рисунку 2.2.

L_1	00	01	10	11
00	00	11	01	10
01	10	01	11	00
10	11	00	10	01
11	01	10	00	11

Рисунок 2.2 – Результат обчислення C1

Далі на рисунку 2.3 наведена квазігрупа L_3, яка використовується для другої операції зашифрування, згідно поточному значенню псевдовипадкової послідовності J_2=”11”.

L_3	00	01	10	11
00	10	01	11	00
01	00	11	01	10
10	01	10	00	11
11	11	00	10	01

Рисунок 2.3 – Таблиця Келі Квазігрупи L3

Для повторення процесу зашифрування при M_2 та P_2, аби знайти C_2, підставимо в вираз (2.3) значення M_2 = “01” та значення P_2 = “10”. При підстановці значень отримуємо C_2=”01”, згідно рис. 2.4

L_3	00	01	10	11
00	10	01	11	0
01	00	11	01	10
10	01	10	00	11
11	11	00	10	01

Рисунок 2.4 – Результат обчислення C_2

Квазігрупа L_2, згідно з основною операцією якої, буде проводитись поточна операція зашифрування, наведена на рисунку 2.5,

L_2	00	01	10	11
00	01	10	00	11
01	11	00	10	01
10	10	01	11	00
11	00	11	01	10

Рисунок 2.5 – Таблиця Келі Квазігрупи L_2

Для отримання C_3, замінимо змінні x та y на змінні M_3 та P_3. Змінна z буде замінена на фрагмент шифротексту C_3. Оскільки M_3=”11”, а P_3=”00”, маємо C_3=”00”. Процес отримання значення C_3 зображено на рисунку 2.6

L_5	00	01	10	11
00	01	10	00	11
01	11	00	10	01
10	10	01	11	00
11	00	11	01	10

Рисунок 2.6 – Результат обчислення C_3

Наведемо квазігрупу L_0, згідно основної операції якої буде проводитись поточна операція. Квазігрупа L_0 наведена на рисунку 2.7

L_0	00	01	10	11
00	11	00	10	01
01	01	10	00	11
10	00	11	01	10
11	10	01	11	00

Рисунок 2.7 – Таблиця Келі Квазігрупи L_0

Підставляючи в формулу (2.3) значення M_4 = ”00” та P_4 = ”11” отримуємо значення C_4= “01”, як показано на рис. 2.8

L_0	00	01	10	11
00	11	00	10	01
01	01	10	00	11
10	00	11	01	10
11	10	01	11	00

Рисунок 2.8 – Результат обчислення C_4

Розшифрування за своїм процесом є аналогічним зашифруванню. В процесі розшифрування застосовується операція лівого ділення:

$$z \cdot y = x$$

Змінна z замінюється на фрагмент шифротексту C_i, змінна y замінюється на фрагмент послідовності P_i, а змінна x замінюється на фрагмент відкритого повідомлення M_i.

Послідовність P та псевдовипадкова послідовність J залишаються незмінними.

Для розшифрування візьмемо квазігрупу з побічного набору, визначену псевдовипадковою послідовністю J. Для першої операції використовується квазігрупа L_1^{-1}. Вказана квазігрупа наведена на рисунку 2.9

L_1^{-1}	00	01	10	11
00	00	10	11	01
01	11	01	00	10
10	01	11	10	00
11	10	00	01	11

Рисунок 2.9 – Таблиця Келі квазігрупи L_1^{-1}

Для отримання M_1, замінимо змінні z та y на змінні C_1 та P_1. Змінна x буде замінена на фрагмент відкритого тексту M_1. Оскільки C_1=”00”, а P_1=”01”, маємо M_1=”10”. Процес отримання значення M_1 зображено на рис. 2.10

L_1^{-1}	00	01	10	11
00	00	10	11	01
01	11	01	00	10
10	01	11	10	00
11	10	00	01	11

Рисунок 2.10 – Результат обчислення M_1

Квазігрупа L_3^{-1}, що використовується для другої операції, наведена в таблиці 2.11:

L_3^{-1}	00	01	10	11
00	01	11	10	00
01	10	00	01	11
10	00	10	11	01
11	11	01	00	10

Рисунок 2.11 – Таблиця Келі квазігрупи L_3^{-1}

Для отримання M_2, замінимо змінні z та y на змінні C_2 та P_2. Змінна x буде замінена на фрагмент відкритого тексту M_2. Оскільки C_2=”01”, а P_2=”10”, маємо M_2=”01”. Процес отримання значення M_2 зображено на рис. 2.12

L_3^{-1}	00	01	10	11
00	01	11	10	00
01	10	00	01	11
10	00	10	11	01
11	11	01	00	10

Рисунок 2.12 – Результат обчислення M_2

Для отримання M_2, замінимо змінні z та y на змінні C_2 та P_2. Змінна x буде замінена на фрагмент відкритого тексту M_2. Оскільки C_2=”01”, а P_2=”10”, маємо M_2=”01”. Процес отримання значення M_2 зображено на рис. 2.12.

Квазігрупа L_5^{-1}, що використовується для третьої операції, наведена в таблиці 2.13:

L_5^{-1}	00	01	10	11
00	11	01	00	10
01	00	10	11	01
10	10	00	01	11
11	01	11	10	00

Рисунок 2.13 – Таблиця Келі Квазігрупи L_5^{-1}

Для отримання M_3, замінимо змінні z та y на змінні C_3 та P_3. Змінна x буде замінена на фрагмент відкритого тексту M_3. Оскільки C_3=”00”, а P_3=”00”, маємо M_3=”11”. Процес отримання значення M_3 зображено на рис. 2.14

L_5^{-1}	00	01	10	11
00	11	01	00	10
01	00	10	11	01
10	10	00	01	11
11	01	11	10	00

Рисунок 2.14 – Результат обчислення M_3

Квазігрупа L_0^{-1}, що використовується для четвертої операції, наведена в таблиці 2.15.

L_0^{-1}	00	01	10	11
00	10	00	01	11
01	01	11	10	00
10	11	01	00	10
11	00	10	11	01

Рисунок 2.15 – Таблиця Келі Квазігрупи L_0^{-1}

Для отримання M_4, замінимо змінні z та y на змінні C_4 та P_4. Змінна x буде замінена на фрагмент відкритого тексту M_4. Оскільки C_4=”01”, а P_4=”11”, маємо M_4=”00”. Процес отримання значення M_3 зображено на рис. 2.16

L_0^{-1}	00	01	10	11
00	10	00	01	11
01	01	11	10	00
10	11	01	00	10
11	00	10	11	01

Рисунок 2.16 – Результат обчислення M_3

Висновки до розділу

В результаті проведеного аналізу запропонованого криптографічного примітиву на основі CIP-квазігруп 4-го порядку з функцією оборотності x^2 серед ізотопів групи Кляйна було визначено, що квазігрупи за своїми властивостями схрещеної оборотності підходять для їх використання в якості криптографічних примітивів для алгоритму потокового шифрування.

Властивість оборотності квазігрупи дозволяє реалізувати процес розшифрування тим самим чином, що й процес за шифрування. А це, в свою чергу, дозволить зменшити апаратні витрати при розробці засобу потокового шифрування на основі середніх CIP-квазігруп з функцією оборотності x^2.

Розроблено метод та описано алгоритм шифрування, що базується на використанні головної операції квазігрупи для зашифрування та операції лівого ділення для розшифрування інформації.

Розроблений алгоритм потокового шифрування складається з семи кроків, двох розгалужень та двох циклів. Цей алгоритм реалізується апаратно за допомогою генератора псевдовипадкової послідовності, регістра зсуву, блоку керування, операційного блоку та комутатора.

Генератор псевдовипадкової послідовності передає псевдовипадкове число до блоку керування, регістр зсуву визначає параметр поточної квазігрупової операції, блок керування визначає режим роботи та поточну квазігрупову операцію на основі псевдовипадкового числа, операційний блок обчислює результат квазігрупової операції, обраної блоком керування, на основі вхідних даних та стану регістра зсуву, а комутатор визначає, які дані надходять на регістр зсуву.

В результаті огляду принципу роботи операційного блоку наведено приклад проведення операції зашифрування та операції розшифрування.

Задачі розділу виконано в повному обсязі.

3 РОЗРОБКА СТРУКТУРИ ТА ОЦІНКА СКЛАДНОСТІ ЗАСОБУ

Третій розділ присвячено розгляду варіантів блоків апаратного засобу потокового шифрування, розробці його структури, оцінці складності розробленого апаратного засобу та порівняльним оцінкам сучасних засобів потокового шифрування з запропонованим засобом.

3.1 Генератор псевдовипадкової послідовності

Для вибору квазігрупи, що використовується в поточній операції, необхідно генерувати псевдовипадкову послідовність. Оскільки було прийняте рішення використовувати два 32-розрядних ключі, постає необхідність в створенні генератора псевдовипадкової послідовності на основі регістра зсуву з ліній ним зворотнім зв’язком розміром в 32 біти.

Генератор псевдовипадкової послідовності є апаратним засобом, який використовується для створення послідовності чисел, що мають властивості, схожі на випадкові числа. Термін "псевдовипадковий" вказує на те, що послідовність, яку створює генератор псевдовипадкових чисел, фактично є детермінованою, а не повністю випадковою.

Генератор псевдовипадкових чисел використовує початкове значення, відоме як "насіння" або "початковий стан", та використовує складні математичні алгоритми для обчислення послідовності чисел. Ці алгоритми зазвичай базуються на математичних формулах або використовують стан попереднього числа для генерації наступного числа. Ключовим аспектом генератора псевдовипадкових чисел є те, що при заданому початковому стані він генерує послідовність чисел, що здається випадковою.

Однак важливо зазначити, що послідовність, згенерована генератором псевдовипадкових чисел, не є повністю випадковою у точному сенсі, оскільки вона є результатом обчислювального процесу і може бути передбаченою або повторюваною, якщо відомі початкові умови та алгоритми генерації.

Двома видами регістрів зсуву, що використовуються в генераторах псевдовипадкових послідовностей є регістр зсуву з лінійним зворотнім зв'язком і регістр зсуву з нелінійним зворотнім зв'язком. Основна різниця між ними полягає в способі визначення зворотного зв'язку між бітами регістру зсуву.

У регістрі зсуву з лінійним зворотнім зв'язком кожен біт регістру зсуву зв'язаний з одним або декількома іншими бітами за допомогою лінійних функцій зворотного зв'язку. Це означає, що вирази, використовувані для визначення зворотного зв'язку, є лінійними комбінаціями (наприклад, XOR-операції) бітів регістру зсуву. Такий підхід дає можливість легко обчислити наступне значення кожного біту на основі попередніх значень.

У регістрі зсуву з нелінійним зворотнім зв'язком зворотний зв'язок між бітами регістру зсуву не є простим лінійним виразом. Замість цього використовуються нелінійні функції зворотного зв'язку, такі як булеві функції зі складними логічними виразами або таблиці заміни. Це дозволяє створювати більш складні і непередбачувані послідовності. Регістри зсуву з нелінійним зворотнім зв'язком можуть бути більш потужними і складними у порівнянні з регістрами зсуву з лінійним зворотнім зв'язком, але вони також вимагають більшої обчислювальної потужності для генерації послідовності.

Таким чином, виходячи з мети роботи, а саме зменшення апаратних витрат засобу потокового шифрування, для створення генератору псевдовипадкової послідовності було обрано саме регістри зсуву з лінійним зворотнім зв’язком.

Розмір регістру зсуву генератора псевдовипадкової послідовності напряму впливає як на криптостійкість алгоритму, за рахунок зменшення або збільшення можливого циклу генератора псевдовипадкової послідовності, так і на апаратну складність засобу шифрування, за рахунок використання більшої або меншої кількості D-тригерів. Оскільки кожен D-тригер потребує 5 елементів для реалізації, їх надмірна кількість може значно збільшити апаратну складність засобу шифрування.

Для використання в засобі потокового шифрування на основі квазігрупи розглядаються три варіанти розміру генератора псевдовипадкової послідовності: 16 бітів, 32 біти та 64 біти. Оскільки важливим є знайти правильний баланс між циклом генератора та складністю його реалізації, розглянемо кожен з варіантів окремо.

Генератор псевдовипадкової послідовності розміром 16 бітів має цикл 65535, що, при використанні двох бітів за раз для вибору квазігрупи буде становити близько 32700 операцій. Припустімо, що один символ повідомлення складається з одного байту, згідно стандарту ACSII [39]. В такому випадку можна зашифрувати близько 8200 символів. Проте надійність такого генератора є сумнівною, тому розглянемо інші варіанти.

Розмір генератора псевдовипадкової послідовності 64 біти є надмірним, адже для його реалізації необхідно 64 D-тригери, а кількість символів, що можна зашифрувати до повторення стану генератора складає $2*10^{18}$. Це набагато більше, ніж необхідно для засобу потокового шифрування, який повинен спеціалізуватись на шифруванні коротких повідомлень.

В результаті стає очевидно, що найкращий баланс між надійністю та простотою реалізації надає саме генератор розміром 32 біти. Кількість можливих станів генератора забезпечує достатню надійність, а складність реалізації 32 тригерів не є надмірною.

В якості полінома для побудови генератора псевдовипадкової послідовності було обрано наступний:

$$X^{32} + X^{30} + X^{26} + x^{25} + 1$$

Цей поліном дає генератору псевдовипадкової послідовності цикл в $2^{32} - 1$ та має мінімальну кількість елементів виняткового АБО в кількості 3.

На рисунку 3.1 зображено схему створеного генератора псевдовипадкової послідовності.

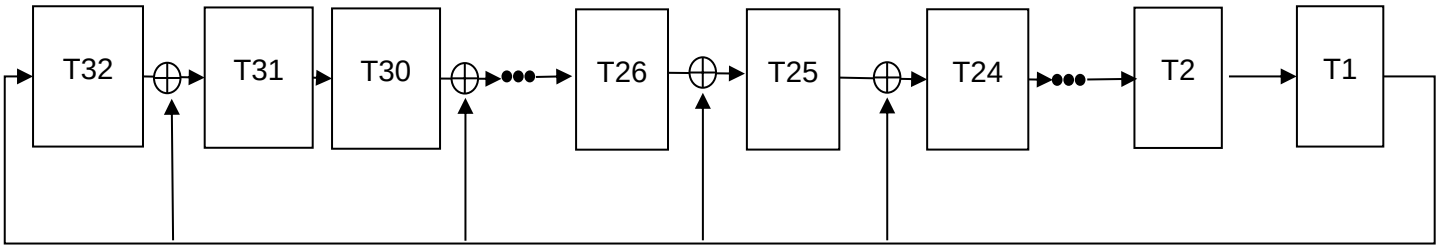


Рисунок 3.1 – Генератор псевдовипадкової послідовності

3.2 Регістр зсуву

Регістр зсуву передає послідовність бітів на вхід операційного блоку. З кожною операцією ключ *K* зсувається на два біти, після чого два біти відкритого повідомлення передаються на регістр зсуву.

Після 16 операцій зашифрування фрагменти повідомлення починають використовуватись в якості послідовності для змінної *Y* в операціях зашифрування або розшифрування.

Подібний підхід позбавляє необхідності в створенні іншого генератора псевдовипадкової послідовності та водночас виключає можливість атаки на цей генератор – без знання повідомлення зловмисник не зможе обійти регістр зсуву, що означає що єдиною ефективною атакою залишається перебір ключа.

Для визначення бітів відкритого повідомлення, що надходять в регістр зсуву необхідно використати комутатор, який, в залежності від керуючого сигналу «зашифрування/розшифрування», буде передавати або вхідні біти відкритого повідомлення, якщо режим роботи є «зашифрування», або вихідні біти відкритого повідомлення, якщо режим роботи є «розшифрування».

На рисунку 3.2 зображено схему регістра зсуву та комутатора

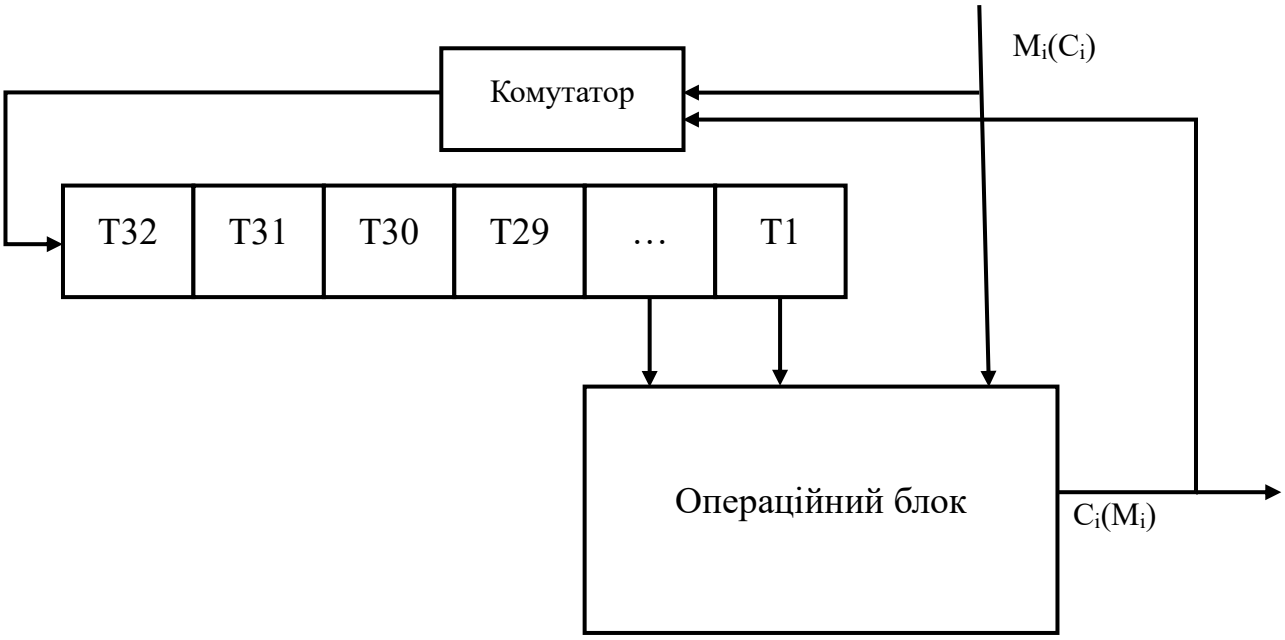


Рисунок 3.2 – Схема регістра зсуву та комутатора

3.3 Блок керування та операційний блок

Блок керування повинен отримувати на вхід керуючий сигнал «зашифрування/розшифрування» та псевдовипадкове число, яке визначає, яка квазігрупа буде використана.

Для визначення квазігрупи, яка використовується, вхідний сигнал від генератора псевдовипадкової послідовності буде направлений на дешифратор. Після цього сигнал дешифратора буде поданий в якості керуючого сигналу до операційного блоку, де за допомогою застосування елементу І до керуючого сигналу та результату роботи операційного блоку для кожної квазігрупи буде визначатись, результат операції якої квазігрупи буде використаний.

Сигнал дешифратора надходить в операційний блок у вигляді шини, від якої керуючі сигнали переходять до модулів конкретних квазігруп, що дозволяє за допомогою двохбітного сигналу обирати, яку з чотирьох квазігруп використовувати для поточної операції.

Керуючий сигнал буде накладено на сигнал дешифратора за допомогою елементів І, визначаючи, який набір квазігруп буде використано: з головною операцією або операцією лівого ділення.

Таким чином, реалізується блок керування, що відповідає за дві функції всередині засобу: вибір режиму роботи на основі контролюючого сигналу «зашифрування/розшифрування» та вибір квазігрупи з конкретного набору для проведення поточної операції.

Операційний блок реалізує 8 квазігруп. З них 4 використовуються для зашифрування, решта – для розшифрування. Після виконання логічної операції І для результату операції кожної квазігрупи з керуючим сигналом з блоку управління, результати цих операцій складаються за логічною операцією АБО. В результаті, на виході операційного блоку отримуємо два біти результату, які, в залежності від режиму роботи, можуть бути шифротекстом C_i або відкритим текстом M_i.

Розглянемо логічні вирази, які описують квазігрупові операції шифрування. Кожна квазігрупа може бути реалізована за допомогою двох логічних виразів. Для початку розглянемо логічні вирази основного набору квазігруп, який використовується при зашифруванні.

В першій частині бакалаврської дипломної роботи встановлено, що квазігрупа L_0 реалізується двома логічними виразами: L_0z_1 та L_0z_2. Для ефективної реалізації квазігрупи в засобі потокового шифрування малоресурсної криптографії необхідно провести спрощення логічних виразів перед їх апаратною реалізацією. Спрощені логічні вирази, що реалізують квазігрупу L_0 мають вигляд:

$$L_0z_1=y_2\overline{(x_1 \oplus x_2)} + \overline{y}_2(x_1 \oplus x_2), \tag{3.1}$$

$$L_0z_2=y_2\overline{(x_1 \oplus y_2)} + \overline{y}_2(x_1 \oplus y_2) \tag{3.2}$$

В результаті можна обчислити кількість логічних елементів, що є необхідними для реалізації цієї квазігрупи в апаратному засобі.

Відповідно до обчислень, які проведені в першій частині комплексної бакалаврської дипломної роботи, маємо 4 елементи виключного АБО, 4 елементи НІ, 4 елементів І та 2 елементи АБО.

Квазігрупа L_1 реалізується двома логічними виразами: L_1z_1 та L_1z_2. Для ефективної реалізації квазігрупи в засобі потокового шифрування малоресурсної криптографії необхідно провести спрощення логічних виразів перед їх апаратною реалізацією. Спрощені логічні вирази, що реалізують квазігрупу L_1:

$$L_1z_1=y_2\overline{(x_1 \oplus x_2)} + \overline{y}_2(x_1 \oplus x_2), \tag{3.3}$$

$$L_1z_2=y_2(x_1 \oplus y_1) + \overline{y}_2\overline{(x_1 \oplus y_1)} \tag{3.4}$$

В результаті обчислено кількість логічних елементів, що є необхідними для реалізації цієї квазігрупи в апаратному засобі. Таким чином, маємо 4 елементи виключного АБО, 4 елементи НІ, 4 елементів І та 2 елементи АБО

Квазігрупа L_2 реалізується двома логічними виразами: L_2z_1 та L_2z_2. Для ефективної реалізації квазігрупи в засобі потокового шифрування малоресурсної криптографії необхідно провести спрощення логічних виразів перед їх апаратною реалізацією. Спрощені логічні вирази, що реалізують квазігрупу L_2:

$$L_2z_1=x_2(x_1 \oplus y_2) + \overline{x}_2\overline{(x_1 \oplus y_2)}, \tag{3.5}$$

$$L_2z_2=y_2(x_1 \oplus y_1) + \overline{y}_2\overline{(x_1 \oplus y_1)} \tag{3.6}$$

В результаті обчислено кількість логічних елементів, що є необхідними для реалізації цієї квазігрупи в апаратному засобі. Таким чином, маємо 4 елементи виключного АБО, 4 елементи НІ, 4 елементів І та 2 елементи АБО.

Квазігрупа L_3 реалізується двома логічними виразами: L_3z_1 та L_3z_2. Для ефективної реалізації квазігрупи в засобі потокового шифрування малоресурсної криптографії необхідно провести спрощення логічних виразів перед їх апаратною реалізацією. Спрощені логічні вирази, що реалізують квазігрупу L_3:

$$L_3z_1=x_2(x_1 \oplus y_2) + \overline{x}_2\overline{(x_1 \oplus y_2)}, \tag{3.3}$$

$$L_3 z_2 = y_2(x_1 \oplus y_1) + \overline{y}_2\overline{(x_1 \oplus y_1)} \quad (3.4)$$

В результаті обчислено кількість логічних елементів, що є необхідними для реалізації цієї квазігрупи в апаратному засобі. Таким чином, маємо 4 елементи виключного АБО, 4 елементи НІ, 4 елементів І та 2 елементи АБО.

Розглянемо логічні вирази, що реалізують квазігрупи з додаткового набору, який використовується для виконання операцій зашифрування. Як і логічні вирази, що реалізують квазігрупи з основного набору, ці логічні вирази також необхідно мінімізувати. Дана мінімізація формул здійснена також в першій частині комплексної БДР.

Квазігрупа L_0^{-1}реалізується двома логічними виразами: $L_0^{-1}z_1$ та $L_0^{-1}z_2$. Для ефективної реалізації квазігрупи в засобі потокового шифрування малоресурсної криптографії необхідно провести спрощення логічних виразів перед їх апаратною реалізацією. Спрощені логічні вирази, що реалізують квазігрупу L_0^{-1}:

$$L_0^{-1}z_1 = x_2\overline{(y_1 \oplus y_2)} + \overline{x}_2(y_1 \oplus y_2), \quad (3.3)$$

$$L_0^{-1}z_2 = x_2\overline{(x_1 \oplus y_1)} + \overline{x}_2(x_1 \oplus y_1) \quad (3.4)$$

В результаті обчислено кількість логічних елементів, що є необхідними для реалізації цієї квазігрупи в апаратному засобі. Таким чином, маємо 4 елементи виключного АБО, 4 елементи НІ, 4 елементів І та 2 елементи АБО.

Квазігрупа L_1^{-1}реалізується двома логічними виразами: $L_1^{-1}z_1$ та $L_1^{-1}z_2$. Для ефективної реалізації квазігрупи в засобі потокового шифрування малоресурсної криптографії необхідно провести спрощення логічних виразів перед їх апаратною реалізацією. Спрощені логічні вирази, що реалізують квазігрупу L_1^{-1}:

$$L_1^{-1}z_1 = x_2(y_1 \oplus y_2) + \overline{x}_2\overline{(y_1 \oplus y_2)}, \quad (3.3)$$

$$L_1^{-1}z_2 = x_1(x_2 \oplus y_1) + \overline{x}_1\overline{(x_2 \oplus y_1)} \quad (3.4)$$

В результаті обчислено кількість логічних елементів, що є необхідними для реалізації цієї квазігрупи в апаратному засобі. Таким чином, маємо 4 елементи виключного АБО, 4 елементи НІ, 4 елементів І та 2 елементи АБО.

Квазігрупа L_2^{-1}реалізується двома логічними виразами: $L_2^{-1}z_1$ та $L_2^{-1}z_2$. Для ефективної реалізації квазігрупи в засобі потокового шифрування малоресурсної криптографії необхідно провести спрощення логічних виразів перед їх апаратною реалізацією. Спрощені логічні вирази, що реалізують квазігрупу L_2^{-1}:

$$L_2^{-1}z_1 = x_2\overline{(y_1 \oplus y_2)} + \overline{x}_2(y_1 \oplus y_2), \tag{3.3}$$

$$L_2^{-1}z_2 = x_1(x_2 \oplus y_1) + \overline{x}_1\overline{(x_2 \oplus y_1)} \tag{3.4}$$

В результаті обчислено кількість логічних елементів, що є необхідними для реалізації цієї квазігрупи в апаратному засобі. Таким чином, маємо 4 елементи виключного АБО, 4 елементи НІ, 4 елементів І та 2 елементи АБО.

Квазігрупа L_3^{-1} реалізується двома логічними виразами: $L_3^{-1}z_1$ та $L_3^{-1}z_2$. Для ефективної реалізації квазігрупи в засобі потокового шифрування малоресурсної криптографії необхідно провести спрощення логічних виразів перед їх апаратною реалізацією. Спрощені логічні вирази, що реалізують квазігрупу L_3^{-1}:

$$L_3^{-1}z_1 = x_2(y_1 \oplus y_2) + \overline{x}_2\overline{(y_1 \oplus y_2)}, \tag{3.3}$$

$$L_3^{-1}z_2 = x_1(x_2 \oplus y_1) + \overline{x}_1\overline{(x_2 \oplus y_1)} \tag{3.4}$$

В результаті обчислено кількість логічних елементів, що є необхідними для реалізації цієї квазігрупи в апаратному засобі. Таким чином, маємо 4 елементи виключного АБО, 4 елементи НІ, 4 елементів І та 2 елементи АБО.

Отримавши мінімізовані логічні функції, що реалізують квазігрупи основного та побічного набору, можна реалізувати операційний блок.

Помітною є тенденція до повторення деяких логічних функцій в наборі зашифрування. Основний набір має всього чотири унікальні логічні вирази, які

описують 8 змінних з чотирьох квазігруп. Перелік логічних виразів, які описують основний набір квазігруп:

$$L_0z_1=L_1z_1=y_2\overline{(x_1 \oplus x_2)} + \overline{y}_2(x_1 \oplus x_2),$$

$$L_0z_2=L_3z_2=y_2\overline{(x_1 \oplus y_2)} + \overline{y}_2(x_1 \oplus y_2),$$

$$L_1z_2=L_3z_2=y_2(x_1 \oplus y_1) + \overline{y}_2\overline{(x_1 \oplus y_1)},$$

$$L_2z_1=L_3z_1=x_2(x_1 \oplus y_2) + \overline{x}_2\overline{(x_1 \oplus y_2)}$$

Як видно з наведених мінімізованих логічних виразів, повторення відбуваються виключно між виразами, які описують одну і ту ж змінну двох квазігруп. Для прикладу, вираз:

$$y_2\overline{(x_1 \oplus y_2)} + \overline{y}_2(x_1 \oplus y_2)$$

описує змінну z_2 квазігруп L_0 та L_3, а вираз:

$$x_2(x_1 \oplus y_2) + \overline{x}_2\overline{(x_1 \oplus y_2)}$$

описує змінну z_1 квазігруп L_2 та L_3. Такий розподіл є доволі зручним для апаратної реалізації, адже не вимагає надмірного використання елементів І з контрольними сигналами.

Тепер розглянемо логічні вирази, які описують додатковий набір квазігруп, що використовується для розшифрування. Як і в основному наборі, в додатковому є всього 4 унікальні логічні вирази, кожен з яких повторюється двічі. Таким чином, вони описують всі 8 змінних серед чотирьох квазігруп. Перелік логічних виразів що реалізують квазігрупи додаткового набору:

$$L_0^{-1}z_1 = L_3^{-1}z_1 = x_2\overline{(y_1 \oplus y_2)} + \overline{x}_2(y_1 \oplus y_2),$$

$$L_3^{-1}z_2 = L_2^{-1}z_2 = x_1(x_2 \oplus y_1) + \overline{x}_1\overline{(x_2 \oplus y_1)},$$

$$L_1^{-1}z_1 = L_2^{-1}z_1 = x_2(y_1 \oplus y_2) + \overline{x}_2\overline{(y_1 \oplus y_2)},$$

$$L_0^{-1}z_2 = L_1^{-1}z_2 = x_2\overline{(x_1 \oplus y_1)} + \overline{x}_2(x_1 \oplus y_1)$$

З наведених мінімізованих логічних виразів видно, що, аналогічно до головного набору, повторювані вирази описують одні і ті ж змінні: два вирази описують z_1, ще два – z_2. Як і у випадку з логічними виразами головного набору, це дозволяє зменшити складність апаратної реалізації за рахунок зменшення кількості елементів І, що необхідні для контролю роботи операційного блоку в режимі розшифрування.

Таким чином, отримано 8 логічних виразів, що необхідно реалізувати в операційному блоці апаратного засобу потокового шифрування. Кожний вираз буде передавати свій результат до логічного елемента І, де буде проводитись операція кон'юнкції з керуючим сигналом, який надходить від блока управління. Далі всі вихідні сигнали, що відповідають за реалізацію однієї змінної, наприклад, z_1, будуть об'єднуватись за допомогою логічної диз'юнкції один з одним. Таким чином, на вихід операційного блоку буде представлено два сигнали: сигнал z_1 та сигнал z_2, які разом становлять результат роботи операційного блоку, та, відповідно, фрагмент шифротексту, або фрагмент відкритого тексту, в залежності від режиму роботи апаратного засобу.

3.4 Розробка структури апаратного засобу потокового шифрування

Для розробки структури апаратного засобу, необхідно визначити, які блоки є необхідними для його побудови. На основі проведеного аналізу алгоритму було визначено такі блоки:

- генератор псевдовипадкової послідовності на основі регістрів зсуву з лінійним зворотнім зв'язком;
- регістр зсуву;
- комутатор;
- блок керування;
- операційний блок;

Генератор псевдовипадкової послідовності було обрано раніше. Розмір генератору становить 32 біти, його твірний поліном: $x^{31} + x^{27} + x^{26} + 1$. Цей генератор послідовності має два виходи, які передають стан двох крайніх регістрів генератора до блоку керування для визначення квазігрупи, що використовується для поточної операції.

Регістр зсуву складається з 32 тригерів, які передають два біти ключа до операційного блоку, після чого відбувається нециклічний зсув, в результаті якого у звільнені комірки записуються два використані біти повідомлення. Ця операція повторюється після кожної ітерації зашифрування або розшифрування, в результаті чого після 16 операцій зашифрування або розшифрування регістр зсуву починає передавати до операційного блоку біти повідомлення.

Комутатор використовується для визначення відкритого повідомлення для подальшої його подачі на вхід регістра зсуву. Від блоку керування до комутатора надходить керуючий сигнал, який, за допомогою двох елементів І та одного елементу АБО визначає, чи будуть використані біти, що надходять на вхід апаратного засобу, у випадку якщо режим роботи є «зашифрування», чи біти, які отримуються на виході операційного блоку, у випадку якщо режим роботи встановлено на «розшифрування».

Блок керування за допомогою дешифратора створює так звану шину, яка дозволяє обирати, яку квазігрупу з операційного блоку використовувати для поточної операції. Вибір режиму здійснюється шляхом створення додаткової шини та використанням елементів І, що проводять операцію кон’юнкції з керівним сигналом та результатом роботи дешифратора. Сигнали з шини надходитимуть до операційного блоку.

Операційний блок працює шляхом виконання логічних виразів, що характеризують квазігрупи основного та побічного набору, та подальших операцій кон'юнкції результатів цих логічних виразів та сигналів з блоку керування. Подальші сигнали за допомогою логічної диз'юнкції об'єднуються для виведення в вигляді двох бітів. Перейдемо до розробки структури програмного засобу.

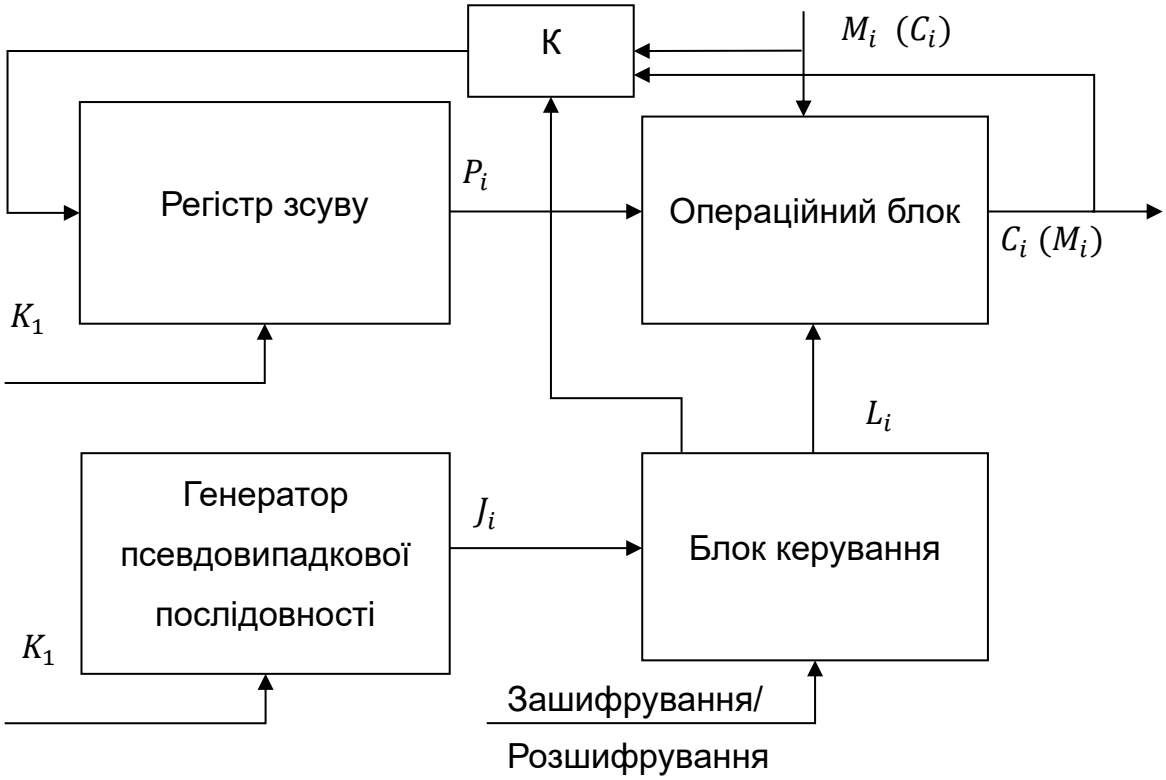


Рисунок 3.3 – Структура засобу потокового шифрування

3.5 Обчислення складності апаратної реалізації

Обчислення складності апаратної реалізації засобу для потокового шифрування виконується за допомогою умовних одиниць GE (Gate Equivalent), які фактично описують кількість логічних вентилів «І-НЕ», які необхідні для реалізації засобу.

Обчислення складності апаратної реалізації засобів відбувається згідно бібліотеки, що містить базові логічні вентилі та таблицю, що описує їх складність в одиницях GE. Для виконання обчислення складності апаратної реалізації засобу для потокового шифрування в даній роботі використовується бібліотека

UMCL18G212T3 [40], що описує основні компоненти, необхідні для виконання задачі. Список компонентів та значення їх апаратної складності в одиницях GE наведено в таблиці 3.1

Таблиця 3.1 – Значення складності в GE для основних елементів апаратної реалізації

Елемент	GE(Gate Equivalent)
НІ	0,67
І	1,33
АБО	1,33
Виключне АБО	2,67
І-НІ	1
АБО-НІ	1
Мультиплексор	2,33
Тригер	5,33

Для обчислення складності апаратної реалізації засобу потокового шифрування необхідно провести підрахунок елементів, необхідних для побудови засобу. Почнемо з регістра зсуву.

Регістр зсуву являє собою послідовний каскад тригерів, які, при отриманні тактового сигналу передають свій стан наступному тригеру, ти самим зсуваючи бітову мапу на один символ. Для реалізації регістру зсуву розміром в 32 біти необхідно 32 D-тригери. Кожен D-тригер має апаратну складність 5,33 GE. Загальна складність реалізації регістра зсуву складає 170,56 GE.

Генератор псевдовипадкової послідовності є схожим до звичайного регістра зсуву, але на додаток ще має декілька елементів «Виключне АБО». Оскільки розмір генератора псевдовипадкової послідовності складає 32 біти, кількість D-тригерів, необхідних для його реалізації, становить 32, що дає загальну складність реалізації регістру зсуву в 170,56 GE. На додаток до регістру зсуву, генератор псевдовипадкової послідовності також містить три елементи «Виключне АБО».

Складність апаратної реалізації елементу «Виключне АБО» становить 2,67 GE. Загальна складність всіх елементів «Виключне АБО» становить 8,01 GE. Тобто, загальна складність генератора псевдовипадкової послідовності на основі регістра зсуву з лінійним зворотнім зв'язком становить 178,57 GE.

Блок керування складається з дешифратора, який будується за допомогою двох елементів «НІ» та чотирьох елементів «І», чотирьох елементів «І» та чотирьох елементів «АБО» для режиму роботи «зашифрування», чотирьох елементів «І» та чотирьох елементів «АБО» для режиму роботи «розшифрування», а також одного елемента «НІ» для обробки контролюючого сигналу «зашифрування/розшифрування». Разом це становить 3 елемента «НІ», 12 елементів «І» та 8 елементів «АБО». Складність апаратної реалізації елемента «НІ» становить 0,67 GE, складність апаратної реалізації елемента «І» становить 1,33 GE, а складність апаратної реалізації елемента «АБО» становить 1,33 GE. Складність реалізації всіх елементів «НІ» становить 2,01 GE, складність реалізації всіх елементів «І» становить 10,64 GE, а складність реалізації всіх елементів «АБО» - 10,64 GE. Таким чином, загальна складність реалізації блоку керування становить 23,29.

Комутатор відповідає за вибір джерела сигналу для регістра зсуву в залежності від режиму роботи та складається з одного елемента «НІ» та чотирьох елементів «І». Загальна складність апаратної реалізації комутатора, згідно даних, наведених в бібліотеці, становить 5,99 GE.

Операційний блок складається з восьми логічних виразів. Кожен з мінімізованих логічних виразів складається з однієї операції «АБО», двох операцій «І», двох операцій «Виняткове АБО» та двох операцій «НІ». Таким чином, апаратна складність реалізації одного виразу становить 10,67 GE. Таким чином, якщо за своєю складністю всі логічні вирази є однаковими, можна визначити, що складність реалізації всіх логічних виразів становить 85,36 GE. Також до операційного блоку належать вісім елементів «І» та шість елементів «АБО», які використовуються для формування результату роботи блоку. Складність реалізації цих елементів

становить 18,62 GE. Таким чином, загальна складність реалізації операційного блоку становить 103,98

Складність окремих модулів засобу потокового шифрування наведено в таблиці 3.2.

Таблиця 3.2 – Складність модулів засобу шифрування

Модуль	Складність (GE)
Генератор псевдовипадкової послідовності	178,57
Регістр зсуву	170,56
Блок керування	23,29
Комутатор	5,99
Операційний блок	103,98

Отримавши значення складності окремих модулів можна обчислити загальну складність реалізації засобу потокового шифрування. В результаті обчислень отримано загальну складність 482,39 GE.

3.6 Порівняльні оцінки

Порівняльні оцінки розробленого апаратного засобу потокового шифрування виконуються шляхом опису основних характеристик запропонованого засобу та одного з сучасних засобів потокового шифрування.

Для початку порівняння візьмемо родину потокових шифрів Grain. Найголовнішою характеристикою порівняння є складність апаратної реалізації засобу. В даному випадку, складність реалізації найпростішого засобу потокового шифру родини Grain становить 1294 GE, в той час як апаратна складність запропонованого засобу потокового шифрування становить всього 482 GE, що є менше майже в 3 рази. Це досягається за рахунок використання змінного набору операцій, що дозволяє зменшити розмір ключа, не жертвуючи криптостікістю.

Засіб шифрування Trivium характеризується, в першу чергу, свою здатністю до паралелізації за рахунок надбудови додаткових регістрів зсуву. З іншого боку, наявність 288 розрядів регістрів зсуву та 1152 таки реалізації є незадовільним показником для алгоритму LW-криптографії, особливо порівняно з запропонованим засобом потокового шифрування, що має всього 2 регістри зсуву, по 32 розряди кожен та довжину процесу ініціалізації в 32 такти. Різниця в складності апаратної реалізації становить приблизно 2000 GE.

Хоча алгоритм потокового шифрування Salsa20 створювався для програмної реалізації, було запропоновано варіант його апаратної реалізації. Цей засіб працює на основі операцій зсуву бітів, додавання 32-бітних слів та операції додавання за модулем 2. Загалом принцип роботи засобу нагадує спрощений варіант алгоритму AES. Складність засобу потокового шифрування становить 3842, що майже в 8 разів більше ніж складність апаратної реалізації запропонованого засобу потокового шифрування на основі CIP-квазігруп

Засіб потокового шифрування Enocoro-80, працюючи на основі п’яти блоків перестановок та двох регістрів зсуву розрядністю 32 біти має надзвичайно високу апаратну складність, що навіть не дозволяє віднести його до засобів LW-криптографії.

Алгоритм F-FCSR-H при аналізі під час конкурсу eSTREAM виявився ненадійним через виявлену атаку. Попри це, складність апаратної реалізації подібного засобу становила 4100 GE, що майже в 10 разів більше ніж апаратна складність реалізованого засобу потокового шифрування на основі CIP-квазігруп 4-го порядку та їх парастрофних перетворень.

Засоби шифрування LIZARD, Fruit-80 та Plantlet принципово мало чим відрізняються від родини шифрів Grain, але їх апаратна складність є набагато меншою. Складність апаратної реалізації цих засобів потокового шифрування становить 1161 GE, 960 GE та 928 GE відповідно. Перевага в складності реалізації запропонованого апаратного засобу потокового шифрування становить від 2 до 2,5 разів.

Алгоритм потокового шифрування EDON-80 використовує набір квазігрупових операцій 4-го порядку для створення гами, яка накладається на повідомлення за допомогою операції додавання за модулем два. Гама генерується за рахунок виконання 80 квазігрупових операцій. В табл. 3.1 наведено результати порівняльних оцінок

Таблиця 3.3 – Порівняльні оцінки сучасних засобів потокового шифрування

Назва алгоритму	Розмір ключа	Тактів ініціалізації	Складність(GE)
Grain	80/128	160	1294/1857
LIZARD	120	128+128	1161
Fruit-80	80	160	960
Plantlet	80	320	928
LILLE	80	720	911
Edon-80	80/128	160	600/1110
Власний (LKRP)	64	32	482,39

В результаті порівняльного аналізу можна зробити висновок, що запропонований засіб потокового шифрування на основі середніх CIP-квазігруп 4-го порядку з функцією оборотності x^2 є надзвичайно простим в реалізації та підходить для використання в LW-криптографії. Складність розробленого засобу для потокового шифрування становить 482,39 GE, що від 1.8 до 3.5 разів менше порівняно із засобами, які реалізують методи потокового шифрування малоресурсної криптографії.

Висновки до розділу

Проаналізувавши отриманий алгоритм роботи засобу потокового шифрування було визначено модулі, що необхідні для побудови розробки структури. До цих модулів належать:

– генератор псевдовипадкової послідовності;

– регістр зсуву;
– блок керування;
– операційний блок;
– комутатор.

Було підібрано конкретні структурні елементи для кожного блоку, розглянуто можливі варіанти деяких блоків. Так, для генератора псевдовипадкової послідовності було обрано розрядність 32 біти та твірний поліном

$$X^{32} + X^{30} + X^{26} + x^{25} + 1$$

Таким чином, генератор має максимальний цикл 2^{32} чисел, що забезпечує оптимальне співвідношення між складністю реалізації та криптостійкістю.

На основі проведеного аналізу та розробленої структури було досліджено складність апаратної реалізації засобу потокового шифрування. Було визначено апаратну складність кожного окремого блоку засобу. Загальна складність засобу потокового шифрування становить 482 GE. Таким чином, розроблений засіб можна застосовувати в малоресурсній криптографії.

В результаті проведеного порівняльного аналізу було визначено, що розроблений засіб потокового шифрування на основі CIP-квазігруп має складність реалізації від 2 до 3.5 разів меншу порівняно з сучасними засобами, що реалізують методи потокового шифрування малоресурсної криптографії.

ВИСНОВКИ

Огляд стану області малоресурсної криптографії показав, що існує необхідність в розробці надлегких засобів в зв’язку з швидким розвитком технологій та тенденцією до мініатюризації апаратних засобів.

Під час дослідження інформаційних джерел щодо засобів потокового шифрування було виявлено закономірність – всі сучасні засоби потокового шифрування, не залежно від того, чи були вони розроблені для використання в малоресурсній криптографії, мають спільну проблему: через використання сталого набору операцій з’являється необхідність в збільшенні розрядності як цих операцій так і секретних ключів та векторів ініціалізації. Це призводить до збільшення складності їх апаратної реалізації за рахунок більшої кількості логічних елементів, зокрема – тригерів для регістрів зсуву.

При огляді прикладів практичного застосування квазігруп в якості криптографічних примітивів виявилось, що застосування квазігруп в якості основи для побудови засобів шифрування є ефективним підходом. Деякі алгоритми шифрування, такі як Edon80 показали свою ефективність на різноманітних конкурсах.

Результат огляду безпосередньо криптографічного примітиву показав, що існує можливість його використання для побудови алгоритму потокового шифрування

В результаті виконання розробки алгоритму шифрування було розроблено метод, який описує процес шифрування та алгоритм роботи засобу, що демонструє процес зашифрування та розшифрування.

На основі отриманого алгоритму було розроблено структуру апаратного засобу, яка складається з генератору псевдовипадкової послідовності, регістру зсуву, блоку керування, операційного блоку та комутатора.

Для реалізації засобу потокового шифрування було обрано генератор псевдовипадкової послідовності, що складається з 32 розрядів та 4 елементів

«Виключне АБО». Запропонований генератор псевдовипадкової послідовності має твірний поліном $X^{32} + X^{30} + X^{26} + x^{25} + 1$ та цикл 2^{32}-1.

Обраний регістр зсуву має 32 розряди та нециклічно зсувається на два біти після кожної операції, отримуючи на вхід два біти відкритого повідомлення. Регістр зсуву передає послідовність *P* на вхід операційного блоку.

Блок керування отримує від генератора псевдовипадкової послідовності поточне значення J_i. На основі цього значення блок керування обирає квазігрупу для використання в поточній операції. Також блок керування отримує керуючий сигнал «зашифрування/розшифрування», на основі якого встановлюється режим роботи засобу потокового шифрування.

Комутатор на основі керуючого сигналу обирає, який сигнал буде передаватись на вхід регістру зсуву.

Операційний блок виконує квазігрупову операцію, що обирається на основі стану блоку керування. В якості вхідних параметрів застосовуються вхідне повідомлення та послідовність *P*. На виході операційного блоку отримуємо повідомлення *M* або шифротекст *C*.

В результаті обчислення апаратної складності реалізації розробленого засобу потокового шифрування було визначено, що його апаратна складність становить 482 GE.

При проведенні порівняльного аналізу, було встановлено, що апаратна складність запропонованого засобу потокового шифрування є на порядок меншою за аналогічні засоби. Для порівняння, складність апаратної реалізації засобу потокового шифрування Grain v1 становить 1294 GE, що майже в три рази більше за розроблений засіб потокового шифрування. Загальна різниця в складності становить від 1.8 до 3,5 разів.

Таким чином, усі сформульовані задачі роботи розв’язано і досягнуто мету дослідження, а саме зменшення апаратних витрат засобу потокового шифрування шляхом розробки алгоритму потокового шифрування на основі CIP-квазігруп 4-го порядку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. AES vulnerabilities. Apache2 ubuntu default page: it works. URL: https://paginas.fe.up.pt/~ei10109/ca/aes-vulnerabilities.html (accessed: 28.05.2023).
2. Cryptography in blockchain - GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/cryptography-in-blockchain/ (accessed: 28.05.2023).
3. Salomon D. Data privacy and security. New York, NY : Springer New York, 2003. URL: https://doi.org/10.1007/978-0-387-21707-9 (accessed: 28.05.2023).
4. A02 cryptographic failures - OWASP top 10:2021. OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. URL: https://owasp.org/Top10/A02_2021-Cryptographic_Failures/ (accessed: 28.05.2023).
5. Varonis. 2019 GLOBAL DATA RISK REPORT FROM THE VARONIS DATA LAB. URL: https://info.varonis.com/hubfs/Varonis%202019%20Global%20Data%20Risk%20Report.pdf. (accessed: 28.05.2023).
6. Cost of a data breach 2022. IBM - Deutschland | IBM. URL: https://www.ibm.com/reports/data-breach (accessed: 28.05.2023).
7. Shimeall T., Spring J. Introduction to information security: a strategic-based approach. Syngress, 2013. 382 p.
8. Goldstine H. H., Goldstine A. The electronic numerical integrator and computer (ENIAC). Mathematical tables and other aids to computation. 1946. Vol. 2, no. 15. P. 97. URL: https://doi.org/10.2307/2002620 (accessed: 09.06.2023).
9. SkyJuice. The TRUTH of TSMC 5nm. Angstronomics | SkyJuice | Substack. URL: https://www.angstronomics.com/p/the-truth-of-tsmc-5nm (accessed: 09.06.2023).
10. Behind The Sound®: how do vacuum tube amplifiers work?. McIntoshLabs. URL: https://www.mcintoshlabs.com/brand/news/Behind-The-Sound---How-do-vacuum-tube-amplifiers-work (accessed: 09.06.2023).

11. Покоління ЕОМ - Все про ЕОМ і комп'ютера. Google Sites. URL: https://sites.google.com/site/vseproeomikomputera/home/pokolinna-eom (дата звернення: 09.06.2023).
12. Submission requirements and evaluation criteria for the lightweight cryptography standardization process. NIST Computer Security Resource Center | CSRC. URL: https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf (accessed: 09.06.2023).
13. ECRYPT noe - preliminary call for stream cipher primitives. ECRYPT. URL: https://www.ecrypt.eu.org/stream/call/ (accessed: 09.06.2023).
14. Hell M., Johansson T., Meier W. Grain: a stream cipher for constrained environments. International journal of wireless and mobile computing. 2007. Vol. 2, no. 1. P. 86. URL: https://doi.org/10.1504/ijwmc.2007.013798 (accessed: 09.06.2023).
15. The eSTREAM portfolio page. ECRYPT. URL: https://www.ecrypt.eu.org/stream/e2-grain.html (accessed: 09.06.2023).
16. Babbage S., Dodd M. The MICKEY stream ciphers. Lecture notes in computer science. Berlin, Heidelberg. P. 191–209. URL: https://doi.org/10.1007/978-3-540-68351-3_15 (accessed: 09.06.2023).
17. Babbage S. The stream cipher MICKEY 2.0. Newbury, 2006. 12 p.
18. De Canni`ere C. Trivium specifications. Katholieke universiteit leuven. P. 2–4. 2006
19. Analysis of area-efficiency vs. unrolling for estream hardware portfolio stream ciphers / F. Alharbi et al. Electronics. 2020. Vol. 9, no. 11. P. 1935. URL: https://doi.org/10.3390/electronics9111935 (accessed: 10.06.2023).
20. M. Hell, T. Johansson, Breaking the F-FCSR-H stream cipher in real time, in Advances in Cryptology— ASIACRYPT 2008. Lecture Notes in Computer Science, vol. 5350/2008 (Springer, Berlin, 2008), pp. 557—569
21. Salsa20/12. ECRYPT. URL: https://www.ecrypt.eu.org/stream/e2-salsa20.html (accessed: 15.06.2023).

22. Hamann M., Krause M., Meier W. LIZARD – A lightweight stream cipher for power-constrained devices. IACR transactions on symmetric cryptology. 2017. P. 45–79. URL: https://doi.org/10.46586/tosc.v2017.i1.45-79 (accessed: 15.06.2023).
23. Fruit-80: a secure ultra-lightweight stream cipher for constrained environments. Entropy. 2018. Vol. 20, no. 3. P. 180. URL: https://doi.org/10.3390/e20030180 (accessed: 15.06.2023).
24. Copeland J., Simpson L. Finding slid pairs for the plantlet stream cipher. ACSW '20: australasian computer science week 2020, Melbourne VIC Australia. New York, NY, USA, 2020. URL: https://doi.org/10.1145/3373017.3373024 (accessed: 15.06.2023).
25. Banik S., Isobe T., Morii M. On design of robust lightweight stream cipher with short internal state. IEICE transactions on fundamentals of electronics, communications and computer sciences. 2018. E101.A, no. 1. P. 99–109. URL: https://doi.org/10.1587/transfun.e101.a.99 (accessed: 15.06.2023).
26. McKay B. D. and Wanless I. M. On the number of Latin Squares // Ann. Combin. − 2005. − No. 9. − P. 335-344.
27. Smith J. D. H. Introduction to Quasigroups and Their Representations. Taylor & Francis Group, 2006. 352 p.
28. Sokhatsky F. M. Parastrophic symmetry in quasigroup theory // VisnykDonetsk national university, Ser. A: natural sciences. — 2016. — No. 1–2. — P. 70–83
29. Шелепало Г. В. Класифікація квазігрупових функційних рівнянь і тотожностей мінімальної довжини : кандидатська дисертація. Хмельницький, 2013. 191 с.
30. Крайнічук Г.В., Пилявець І.Ю., Радченко Є.В. CIP-квазігрупи 4-го порядку з оборотним елементом X2 серед ізотопів групи Клейна // Збірник наук. праць Міжн. наук. конф. «Сучасні проблеми мех. та мат. – 2023», ІПММ ім. Я.С.Підстригача. Львів. 2023. – С. 285-286.
31. Shcherbacov V. Elements of quasigroup theory. Elements of quasigroup theory and applications. 2017. P. 3–106. URL: https://doi.org/10.1201/9781315120058-2 (accessed: 15.06.2023).

32. Fawaz M., Zorkta H., Alnazer S. A public key cipher algorithm based on multivariate cubic quasigroups (MCQ). The international conference on electrical engineering. 2010. Т. 7, № 7. С. 1–11. URL: https://doi.org/10.21608/iceeng.2010.33467 (accessed: 28.05.2023).
33. Gligoroski D. Stream cipher based on quasigroup string transformations. м. Skopje, 22 квіт. 2004 р. Skopje, 2004. С. 7–12.
34. 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 (accessed: 31.05.2023).
35. 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-note-on-the-Edon-80-S-box-Bjørstad/fe7114a2ae9c605e026a5cfd61758f3d303f26f6 (accessed: 31.05.2023).
36. Wang X., Xu Y. Modification of Edon80 to Resist the Key Recovery Attack. First International Conference on Information Science and Electronic Technology (ISET 2015), м. Wuhang, China, 21–22 берез. 2015 р. Paris, France, 2015. URL: https://doi.org/10.2991/iset-15.2015.2 (accessed: 31.05.2023).
37. Крайнічук Г.В., Радченко Є.В., Пилявець І.Ю. Концепція шифру на основі CIP-квазігруп // Матеріали Всеукр. наук.-практ. конф. «Theoretical and Applied Cybersecurity (TACS-2023)», присвяч. 100-річному ювілею акад. В. М. Глушкова, КПІ ім. Ігоря Сікорського. – Київ. 2023. – 3 С.
38. Лужецький В.А., Крайнічук Г.В., Радченко Є.В., Пилявець І.Ю. алгоритм шифрування на основі CIP-квазігруп // Матеріали тезів IX Міжн. наук.-техн. конф. «ЗАХИСТ ІНФОРМАЦІЇ І БЕЗПЕКА ІНФОРМАЦІЙНИХ СИСТЕМ», НУ «Львівська політехніка». Львів. 2023. – С.
39. Таблиця символів ASCII - Підтримка від Microsoft. Microsoft Support. URL: https://support.microsoft.com/uk-ua/office/таблиця-символів-ascii-d13f58d3-7bcb-44a7-a4d5-972ee12e50e0 (дата звернення: 16.06.2023).

40. UMCL18G212T3. Virtual Silicon standard cell library based on the UMC L180 0.18 μm 1P6M Logic process. Вид. офіц. 2004. 1 с.

ДОДАТКИ

Додаток А
ПЕРЕЛІК ПУБЛІКАЦІЙ ЗА ТЕМОЮ

1) Крайнічук Г.В., Пилявець І.Ю., Радченко Є.В. CIP-квазігрупи 4-го порядку з оборотним елементом X2 серед ізотопів групи Клейна // Збірник наук. праць Міжн. наук. конф. «Сучасні проблеми мех. та мат. – 2023», ІПММ ім. Я.С.Підстригача. Львів. 2023. – С. 285-286.
2) Лужецький В.А., Крайнічук Г.В., Радченко Є.В., Пилявець І.Ю. Алгоритм шифрування на основі CIP-квазігруп // Матеріали тезів IX Міжн. наук.-техн. конф. «ЗАХИСТ ІНФОРМАЦІЇ І БЕЗПЕКА ІНФОРМАЦІЙНИХ СИСТЕМ», НУ «Львівська політехніка». Львів. 2023. – С.
3) Крайнічук Г.В., Радченко Є.В., Пилявець І.Ю. Концепція шифру на основі CIP-квазігруп // Матеріали Всеукр. наук.-практ. конф. «Theoretical and Applied Cybersecurity (TACS-2023)», присвяч. 100-річному ювілею акад. В. М. Глушкова, КПІ ім. Ігоря Сікорського. Київ. 2023. – 3 С.
4) Крайнічук Г.В., Пилявець І.Ю., Радченко Є.В. Огляд методів шифрування за допомогою квазігрупових операцій // Матеріали тезів LII наук.-техн. конф. ФІТКІ, ВНТУ. Вінниця. 2023 – 2 С.

Додаток Б

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ

1) Міжнародна наукова конференція «Сучасні проблеми механіки та математики – 2023», ІПММ ім. Я.С.Підстригача. Львів. 24-26 травня 2023. – С. 285-286.
2) IX Міжнародна науково-технічна конференція «ЗАХИСТ ІНФОРМАЦІЇ І БЕЗПЕКА ІНФОРМАЦІЙНИХ СИСТЕМ», Національний Університет «Львівська політехніка». Львів 2023.
3) Всеукраїнська науково-практична конференція «Theoretical and Applied Cybersecurity (TACS-2023)», присвячена 100-річному ювілею академіка В. М. Глушкова, Київський Політехнічний Інститут ім. Ігоря Сікорського. Київ 2023
4) LII науково-технічна конференція Факультету інформаційних технологій та комп’ютерної інженерії, Вінницький Національний Технічний Університет. Вінниця, 21-23 червня 2023

Додаток В
ПРОТОКОЛ ПЕРЕВІРКИ БАКАЛАВРСЬКОЇ ДИПЛОМНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Засіб шифрування даних на основі квазігрупи. Частина 2. Криптографічний алгоритм

Автор роботи: Радченко Євгеній Валентинович

Тип роботи: бакалаврська дипломна робота (БДР, МКР).

Підрозділ кафедра захисту інформації ФІТКІ (кафедра, факультет).

Показники звіту подібності Unicheck

Оригінальність – 93,2%. Схожість – 6,8%.

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 виконання автором. Роботу направити на розгляд експертної комісії кафедри.

□ 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.

Особа, відповідальна за перевірку _____ (підпис) Каплун В. А. (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ (підпис) Радченко Є.В. (прізвище, ініціали)

Керівник роботи _____ (підпис) Шелехало І.В. (прізвище, ініціали)

ІЛЮСТРАТИВНА ЧАСТИНА

Засіб шифрування даних на основі квазігрупи. Частина 2. Криптографічний алгоритм

Виконав: студент 4 курсу групи 1БС-19б
спеціальності 125 Кібербезпека
_____ Радченко Є. В.
17 червня 2023 р.
Керівник: к.ф.-м.н., доцент каф. ЗІ
_____ Шелепало Г. В.
17 червня 2023 р.

Таблиці Келі парастрофних квазігрупових операцій

Основний набір

0)	0	1	2	3
0	0	3	1	2
1	2	1	3	0
2	3	0	2	1
3	1	2	0	3

1)	0	1	2	3
0	1	2	0	3
1	3	0	2	1
2	2	1	3	0
3	0	3	1	2

3)	0	1	2	3
0	2	1	3	0
1	0	3	1	2
2	1	2	0	3
3	3	0	2	1

5)	0	1	2	3
0	3	0	2	1
1	1	2	0	3
2	0	3	1	2
3	2	1	3	0

Побічний набір

7)	0	1	2	3
0	0	2	3	1
1	3	1	0	2
2	1	3	2	0
3	2	0	1	3

2)	0	1	2	3
0	1	3	2	0
1	2	0	1	3
2	0	2	3	1
3	3	1	0	2

4)	0	1	2	3
0	2	0	1	3
1	1	3	2	0
2	3	1	0	2
3	0	2	3	1

6)	0	1	2	3
0	3	1	0	2
1	0	2	3	1
2	2	0	1	3
3	1	3	2	0

Таблиця Келі групи Кляйна

	0	**1**	**2**	**3**
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

Список середніх CIP-квазігруп

1)	0	1	2	3
0	1	2	0	3
1	3	0	2	1
2	2	1	3	0
3	0	3	1	2

3)	0	1	2	3
0	2	1	3	0
1	0	3	1	2
2	1	2	0	3
3	3	0	2	1

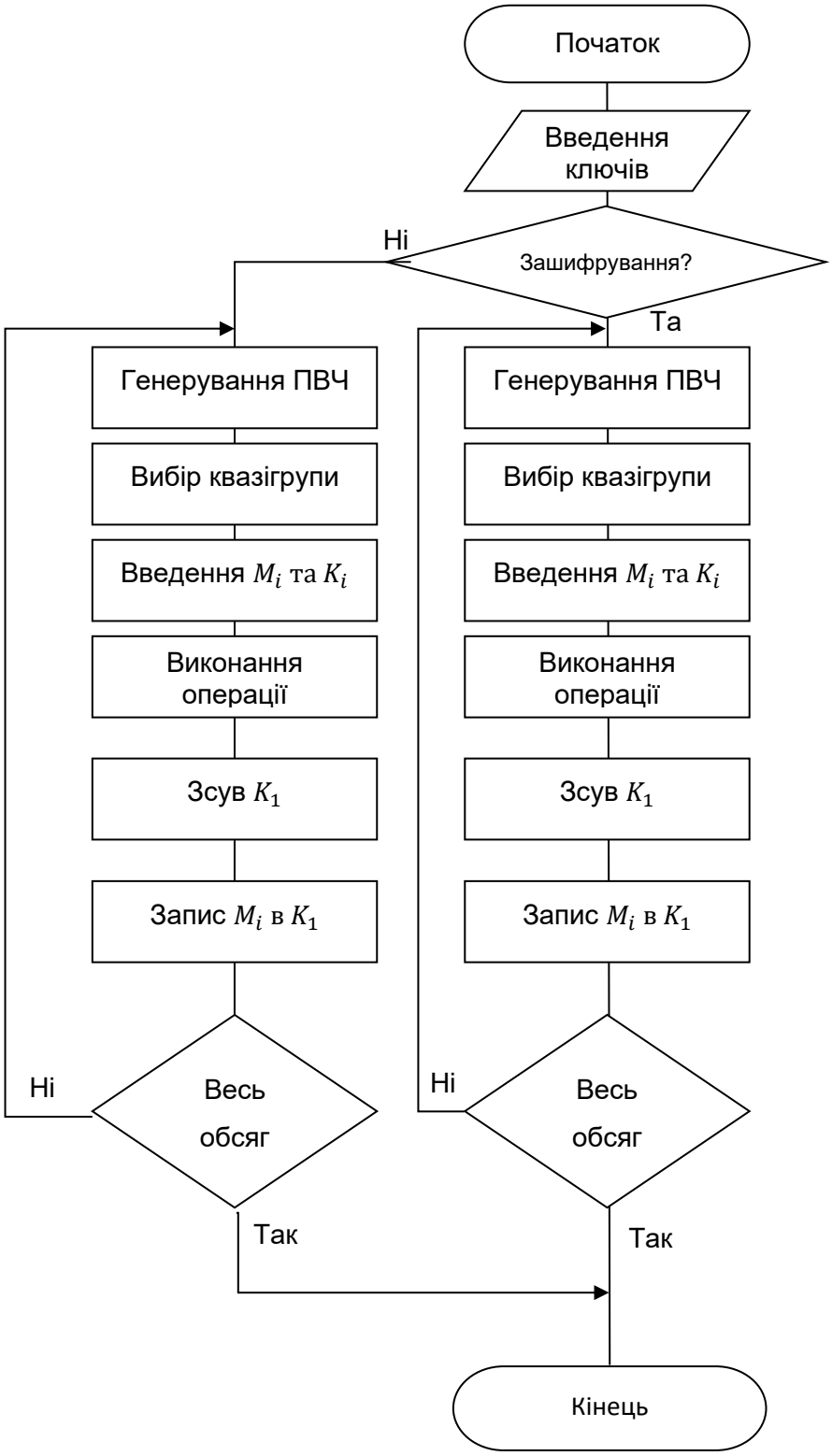
5)	0	1	2	3
0	3	0	2	1
1	1	2	0	3
2	0	3	1	2
3	2	1	3	0

2)	0	1	2	3
0	1	3	2	0
1	2	0	1	3
2	0	2	3	1
3	3	1	0	2

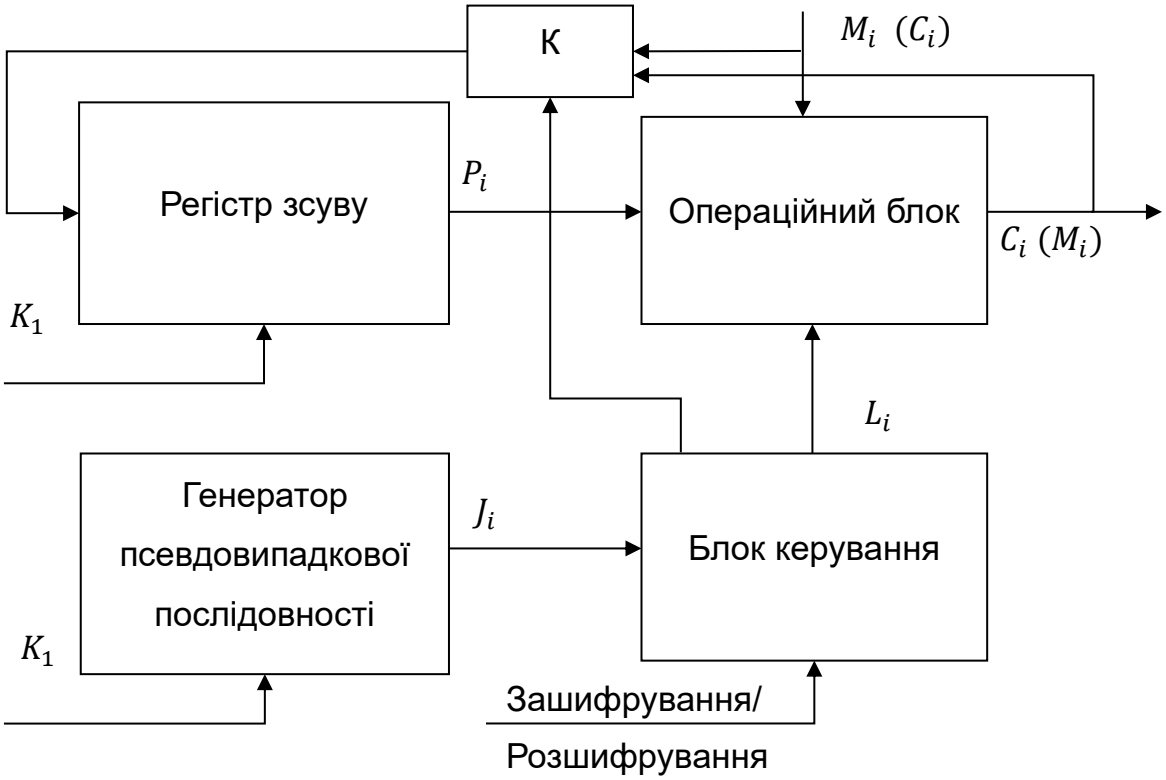
4)	0	1	2	3
0	2	0	1	3
1	1	3	2	0
2	3	1	0	2
3	0	2	3	1

6)	0	1	2	3
0	3	1	0	2
1	0	2	3	1
2	2	0	1	3
3	1	3	2	0

Схема алгоритму шифрування



Структура засобу шифрування



Таблиця з оцінками складності апаратної реалізації засобу

Елемент	GE(Gate Equivalent)
НІ	0,67
І	1,33
АБО	1,33
Виключне АБО	2,67
І-НІ	1
АБО-НІ	1
Мультиплексор	2,33
Тригер	5,33