НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
Навчально-науковий фізико-технічний інститут
Кафедра математичних методів захисту інформації

«На правах рукопису»
УДК 004.056.53

«До захисту допущено»
Зав. кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2024 р.

Дипломна робота

на здобуття ступеня магістра
за освітньо-професійною програмою
«Математичні методи криптографічного захисту інформації»

зі спеціальності: 113 Прикладна математика
на тему: **«Криптографічні властивості множення за модулем та їх застосування у криптоаналізі блокових шифрів»**

Виконав:
студент II курсу, групи ФІ-22мн
Паршин Олександр Юрійович _____

Керівник:
зав. каф. ММЗІ. к.т.н.
Яковлєв Сергій Володимирович _____

Рецензент:
доцент каф. ММАД, к.т.н.
Лавренюк Алла Миколаївна _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Паршин О. Ю. _____

Київ — 2024

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
Навчально-науковий фізико-технічний інститут
Кафедра математичних методів захисту інформації

Рівень вищої освіти — другий (магістерський)
Спеціальність — 113 Прикладна математика,
ОПП «Математичні методи криптографічного захисту інформації»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2024 р.

ЗАВДАННЯ
на магістерську дисертацію

Студент: Паршин Олександр Юрійович

1. Тема роботи: «*Криптографічні властивості множення за модулем та їх застосування у криптоаналізі блокових шифрів*», науковий керівник дисертації: зав. каф. ММЗІ. к.т.н. Яковлєв Сергій Володимирович,

затверджені наказом по університету №__ від «__» _____ 2024 р.

2. Термін подання студентом роботи: «__» _____ 2024 р.

3. Об'єкт дослідження: Конструктивні елементи блокових шифрів на основі схеми Лая-Мессі.

4. Предмет дослідження: Використання алгебраїчних операцій в ключових суматорах шифрів на основі схеми Лая-Мессі.

5. Перелік завдань:

1) Огляд та порівняння ключових суматорів блокових шифрів на основі схеми Лая-Мессі;

2) Пропозиція теоретичної диференціальної атаки на основі ключового суматора на шифр IDEA;

3) Введення нової операції для удосконалення існуючого ключового суматора шифру IDEA;

4) Порівняння складності проведення запропонованої атаки на оригінальний шифр IDEA та з модифікованими ключовими суматорами.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: Презентація доповіді.

7. Орієнтовний перелік публікацій: доповідь на II Всеукраїнській конференції з питань теоретичної і прикладної кібербезпеки «Theoretical and Applied Cybersecurity - TACS-2024» (30 – 31 травня 2024 р., м. Київ, Україна).

8. Дата видачі завдання: 10 вересня 2023 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	01-15 вересня 2023 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	Вересень-жовтень 2023 р.	Виконано
3	Проведення порівняльного аналізу способів захисту схеми Лая-Мессі від структурної слабкості	Листопад-грудень 2023 р.	Виконано
4	Аналіз диференціальних властивостей запропонованої операції $\otimes$	Січень-лютий 2024 р.	Виконано
5	Формулювання диф. атаки на шифр IDEA та обрахунок диференціальних характеристик	Березень-квітень 2024 р.	Виконано
6	Опис та оформлення отриманих результатів	Травень 2024 р.	Виконано

Студент _____ Паршин О.Ю.

Керівник _____ Яковлєв С.В.

РЕФЕРАТ

Кваліфікаційна робота містить: 45 стор., 6 рисунків, 4 таблиці, 15 джерел.

Метою даної роботи є покращення стійкості блокових шифрів на основі схеми Лая-Мессі для забезпечення конфіденційності інформації.

Об'єктом дослідження виступають конструктивні елементи блокових шифрів на основі схеми Лая-Мессі.

Предмет дослідження – використання алгебраїчних операцій в ключових суматорах шифрів на основі схеми Лая-Мессі.

В даній роботі оглянуто використання алгебраїчних операцій в блокових шифрах, побудованих на основі схеми Лая-Мессі. Також наведено нову, теоретичну диференціальну атаку на шифр IDEA. Одержано оцінки стійкості даного шифру до атак подібного типу, а також запропоновано зміну в дизайні ключового суматора, яка дозволяє підвищити стійкість IDEA до диференціального криптоаналізу.

БЛОКОВИЙ ШИФР; СХЕМА ЛАЯ-МЕССІ; IDEA; ДИФЕРЕНЦІАЛЬНИЙ КРИПТОАНАЛІЗ.

ABSTRACT

Qualification work contains: 45 pp., 6 figures, 4 tables, 15 sources.

The goal of this work is to improve the robustness of block ciphers based on the Lai-Massey scheme to ensure information confidentiality.

The object of the research is the constructive elements of block ciphers based on the Lai-Massey scheme.

The subject of the research is the use of algebraic operations in the key adders of block ciphers based on the Lai-Massey scheme.

This work reviews the use of algebraic operations in block ciphers built on the basis of the Lai-Massey scheme. Additionally, a new, theoretical differential attack on the IDEA cipher is presented. The robustness of this cipher against such types of attacks is assessed, and a modification in the design of the key adder is proposed, which enhances the resistance of IDEA to differential cryptanalysis.

BLOCK CIPHER; LAI-MASSEY SCHEME; IDEA; DIFFERENTIAL CRYPTANALYSIS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

$\oplus$ — операція побітового додавання

$\boxplus$ — операція модульного додавання за модулем 2^w

$\odot$ — операція множення за модулем $2^w + 1$, з умовою $0 \equiv 2^w$

$\boxminus$ — операція, обернена до $\boxplus$

$\boxdot$ — операція, обернена до $\odot$

$\otimes$ — операція, визначена як $x \otimes y = (x \boxplus 1) \cdot (y \boxplus 1) \boxminus 1$, де $\cdot$ - множення за модулем $2^w + 1$

ВСТУП

Актуальність дослідження. Більшість блокових шифрів на основі схеми Лая-Мессі хоч і були запропоновані близько 20 років тому, але і зараз залишаються стійкими до відомих методів криптоаналізу. Вони мають певні переваги над блоковими шифрами, заснованими на основі схемі Фейстеля або SP-мереж, зокрема, мають меншу кількість раундів та використовують зручні алгебраїчні операції, тому їх дослідження залишається актуальним.

Метою дослідження в даній роботі є покращення стійкості блокових шифрів на основі схеми Лая-Мессі для забезпечення конфіденційності інформації. Одним зі способів її часткового досягнення є розв'язок наступних задач:

1) провести огляд опублікованих джерел за тематикою дослідження і структурувати загальновідому інформацію;

2) провести порівняльний аналіз способів усунення структурної особливості схеми Лая-Мессі;

3) запропонувати нові способи атак на існуючі блокові шифри на основі схеми Лая-Мессі;

4) означити складність запропонованих атак та їх теоретичний або практичний характер.

Об'єктом дослідження виступають конструктивні елементи блокових шифрів на основі схеми Лая-Мессі.

Предметом дослідження є використання алгебраїчних операцій в ключових суматорах шифрів на основі схеми Лая-Мессі.

При розв'язанні поставлених завдань використовувались такі **методи дослідження**:

1) методи лінійної та абстрактної алгебри;

2) методи диференціального криптоаналізу;

3) методи комбінаторного аналізу;

4) методи комп'ютерного моделювання.

Наукова новизна даного дослідження полягає в пропозиції теоретичної диференціальної атаки нового типу на шифри PES, IDEA, MESH-64, MESH-96, MESH-128, а також визначенні оцінок стійкості даних шифрів від запропонованої атаки.

Практичне значення результатів полягає в пропозиції ускладнення ключових суматорів вищеописаних шифрів з метою підвищити їхню структурну стійкість до наведеного типу атак.

Апробація результатів та публікації. Результати даної роботи були представлені на II Всеукраїнській конференції з питань теоретичної і прикладної кібербезпеки «Theoretical and Applied Cybersecurity - TACS-2024» (30 – 31 травня 2024 р., м. Київ, Україна).

1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО СХЕМУ ЛАЯ-МЕССІ ТА БЛОКОВІ ШИФРИ НА ЇЇ ОСНОВІ

Схема Лая-Мессі (ЛМ) є криптографічною структурою, яка використовується при побудові певних блокових шифрів. Вона є менш розповсюдженою, ніж схема Фейстеля або SP-мережа, але при цьому має ряд переваг і вносить певну різноманітність в дизайн і побудову криптопримітивів.

1.1 Опис схеми Лая-Мессі та приклади шифрів на її основі

Перш ніж почати опис схеми Лая-Мессі та її потенційних слабкостей, необхідно згадати схему Фейстеля, яка є більш вживаним рішенням при побудові блокових шифрів. Це зумовлено її простотою та відсутністю очевидних залежностей, які можна було б використати при криптоаналізі блокових шифрів. Ця схема виглядає наступним чином: $Feist(x, y) = (y, x + F(y))$.

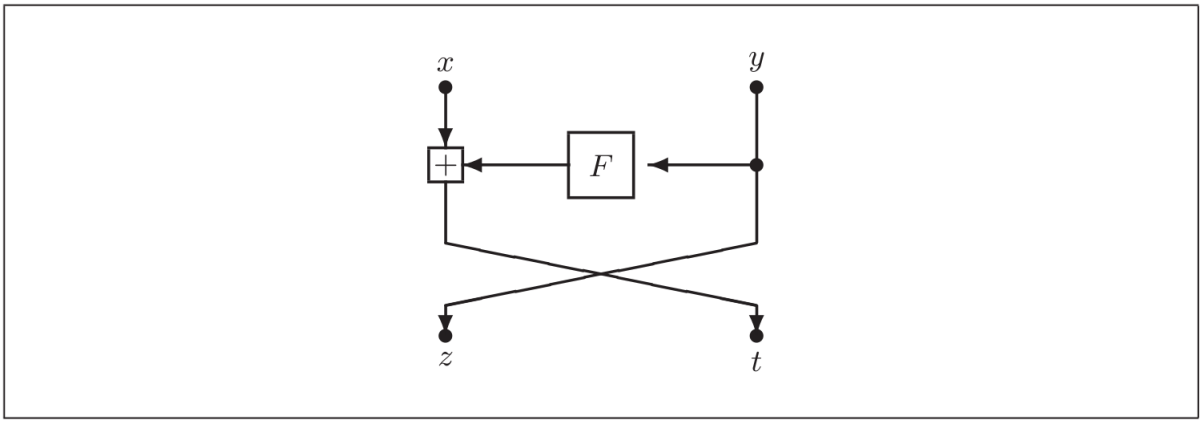

Рисунок 1.1 – Схема Фейстеля

Схема Лая-Мессі має іншу будову: $LM(x, y) = (x + F(x - y), y + F(x - y)) = (z, t)$. Одразу треба зазначити,

що в неї є структурна особливість: $z - t = x - y$. Подібна слабкість вимагає додаткових маніпуляцій для того, щоб схема була стійкою.

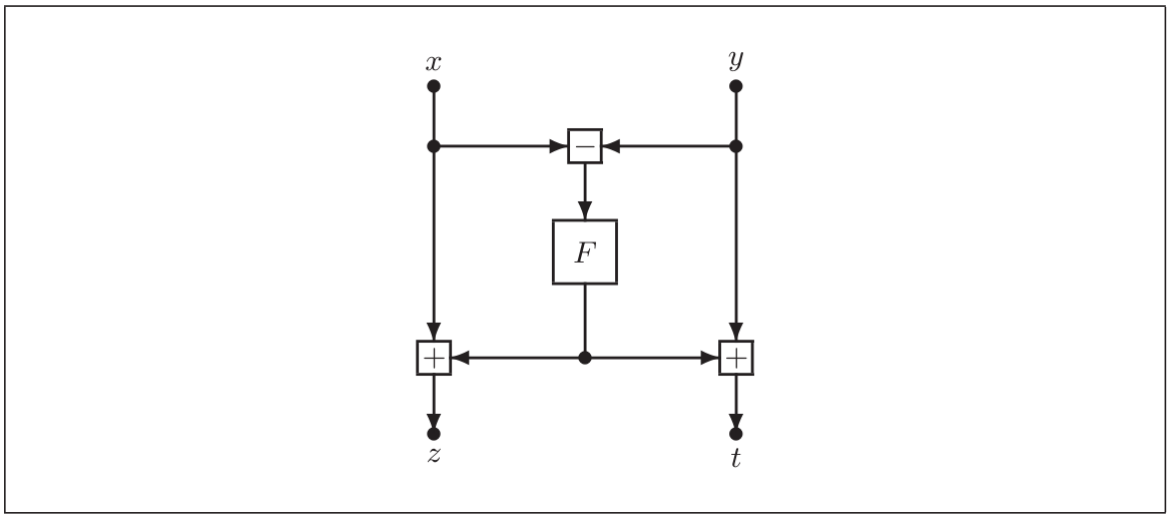

Рисунок 1.2 – Базова схема Лая-Мессі

Але, не дивлячись на таку властивість, існує достатньо велика категорія блокових шифрів, які засновані саме на схемі Лая-Мессі, оскільки вона пропонує і певні переваги, як-от:

1) Раундові функції забезпечують повне перемішування вхідного тексту за один раунд шифрування;

2) Використання меншої кількості раундів шифрування, порівняно з схемою Фейстеля та SP-мережею;

3) Відсутність необхідності використання S-блоків або MDS-матриць;

4) Використання ефективно-обчислювальних алгебраїчних операцій для шифрування.

Також в роботі [6] було показано наведено структуру узагальненої схеми Лая-Мессі, та доведена її нееквівалентність узагальненій схемі Фейстеля. Відповідно, схема Лая-Мессі є окремим структурним дизайном, який може привнести різноманіття в сімейства блокових шифрів.

Приклади шифри, які побудовані на основі схеми Лая-Мессі, наведені у таблиці 1.1.

Таблиця 1.1 – Шифри на основі схеми Лая-Мессі

Шифр	Розмір блоку (бітів)	Розмір ключа (бітів)	Кількість раундів	Розмір слова (бітів)	Рік
PES	64	128	8.5	16	1990
IDEA	64	128	8.5	16	1991
YBC	64	128	6.5	16	1996
RIDEA	64	128	8.5	16	2002
MESH-64	64	128	8.5	16	2003
MESH-96	96	192	10.5	16	2003
MESH-128	128	256	12.5	16	2003
FOX-64	64	[0, 256]	[12, 255]	8	2004
FOX-128	128	[0, 256]	[12, 255]	8	2004
MESH-64(8)	64	128	8.5	8	2005
MESH-128(8)	128	256	8.5	8	2005
Bel-T	128	128, 192, 256	8	32	2007
WIDEA-4	256	512	8.5	16	2009
WIDEA-8	512	1024	8.5	16	2009
IDEA*	64	128	6.5	16	2013
REESSE-3	128	256	8.5	16	2014

1.2 Ліквідація структурної слабкості схеми Лая-Мессі за допомогою ортоморфізмів

Одним зі способів вдосконалення схеми Лая-Мессі є внесення в її структуру ортоморфного відображення. Необхідність і достатність такої зміни для стійкості схеми були показані у роботі [13]. Модифікована схема була запропонована для використання у шифрі WALNUT в [12]. Але даний шифр є скоріше виключенням з родини сімейства шифрів, заснованих на схемі Лая-Мессі, оскільки шифри родини -NUT (COCONUT, PEANUT, DONUT та інші) будувалися на основі схеми Фейстеля, а WALNUT продемонстрував можливість використання і схеми Лая-Мессі.

Означення 1.1. Нехай $\langle G, \cdot \rangle$ — група порядку n. Тоді: $\phi : G \to G$ — ортоморфізм, якщо $\phi(x)$ — бієкція і $\theta(x) = \phi(x) \cdot x^{-1}$ — теж бієкція.

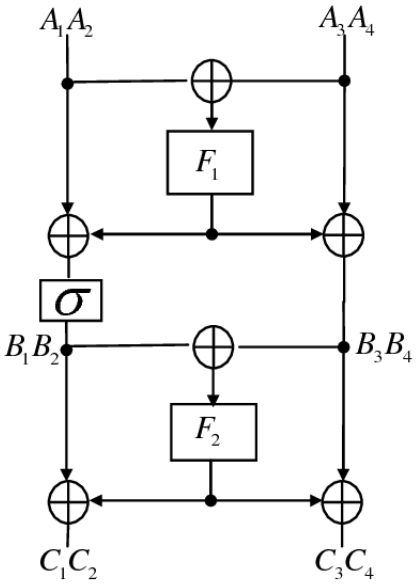

Рисунок 1.3 – Схема Лая-Мессі з ортоморфним перетворенням

Але з даним підходом пов'язано декілька проблем. По-перше, не для будь-якої групи можлива побудова ортоморфізму. Згідно до роботи [7], такі відображення існують над скінченними абелевими групами тоді і тільки тоді, коли їхній порядок є напарним або $\mathbb{Z}_2^2$ ізоморфна одній з їхніх підгруп. Наприклад, групи вигляду $\mathbb{Z}_{2^m}$ не мають ортоморфізмів.

По-друге, постає питання про ефективну генерацію подібних відображень та ефективність іхнього використання. В роботі [5] було наведено декілька подібних алгоритмів, але вони не пов'язані зі схемою Лая-Мессі та не адаптовані для використання на групах, які можна було б використовувати при побудові блокових шифрів.

Таким чином, подібні варіації схеми Лая-Мессі не набули широкого розповсюдження. Більш плідним виявився інакший підхід, заснований на поєднанні декількох алгебраїчних операцій. Переважна більшість шифрів (усі, наведені в таблиці 1.1) використовують саме їх для руйнування структурної особливості схеми Лая-Мессі. В роботі [11] наводиться приклад схеми Лая-Мессі з ключовим суматором - це видно на рисунку 1.4.

Як видно на рисунку 1.4, схема зазнала певнихз змін. По-перше, це

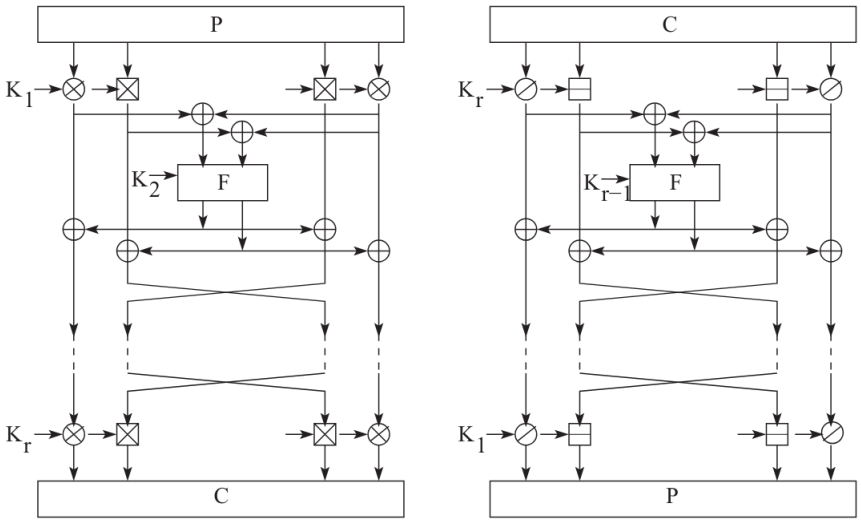

Рисунок 1.4 – Схема Лая-Мессі з ключовим суматором

її паралелізація - в модифікованому варіанті результати роботи двох схем поєднуються між собою. В сучасніших схемах (MESH-96, MESH-128, WIDEA тощо) схема може мати декілька блоків, які можуть обчислюватись паралельно.

По-друге, додано ключовий суматор - перед тим, як бути поданими в схему, входи певним чином замішуються з ключем. Саме це перетворення унеможливлює використання структурної схеми Лая-Мессі. Відповідно, для того, щоб гарантувати стійкість, необхідно, щоб ключовий суматор руйнував залежності між входами якнайшвидше.

1.3 Операції, що використовуються в шифрах родини PES

Шифрами родини PES можна називати усі шифри, наведені в таблиці 1.1. PES (Proposed Encryption Standard) є першим шифром, побудованим на схемі Лая-Мессі з використанням ключового суматора, і всі його подальші нащадки використовували операції, запропоновані в специфікації, з певними модифікаціями.

Цими операціями є додавання за модулем 2^w - $\boxplus$, множення за модулем $2^w + 1$ - $\odot$, та побітове додавання - $\oplus$.

Зауваження. В операції $\odot$ є виключення: $2^w \equiv 0$.

Дана вимога зумовлена декількома причинами:

1) Без цієї зміни елемент 0 в $GF(2^w + 1)$ не матиме оберненого елемента, а з нею він стає оберненим самим до себе.

2) Без цієї зміни елемент 2^w буде єдиним елементом, що не буде вкладатись в w бітів. Отже, з цією зміною програмна реалізація шифру є простішою і ефективнішою.

Вибір операцій для використання також зумовлений рядом вимог [11]:

1) Несумісність операцій: вони не мають мати властивостей дистрибутивності та асоціативності в загальному випадку: $\exists\ a, b, c \in \mathbb{Z}_2^w$:

– $a \boxplus (b \oplus c) \neq (a \boxplus b) \oplus (a \boxplus c)$

– $a \boxplus (b \odot c) \neq (a \boxplus b) \odot (a \boxplus c)$

– $a \oplus (b \boxplus c) \neq (a \oplus b) \boxplus (a \oplus c)$

– $a \oplus (b \odot c) \neq (a \oplus b) \odot (a \oplus c)$

– $a \odot (b \oplus c) \neq (a \odot b) \oplus (a \odot c)$

– $a \odot (b \boxplus c) \neq (a \odot b) \boxplus (a \odot c)$

– $a \boxplus (b \oplus c) \neq (a \boxplus b) \oplus c$

– $a \boxplus (b \odot c) \neq (a \boxplus b) \odot c$

– $a \oplus (b \boxplus c) \neq (a \oplus b) \boxplus c$

– $a \oplus (b \odot c) \neq (a \oplus b) \odot c$

– $a \odot (b \oplus c) \neq (a \odot b) \oplus c$

– $a \odot (b \boxplus c) \neq (a \odot b) \boxplus c$

2) Жодна операція не повторюється двічі підряд впродовж шифрування або розшифрування (це можна побачити на малюнку 2.1).

3) Кожна з операцій необхідна:

– Відкидання операції $\oplus$ робить шифр афінним над $(\mathbb{Z}_{2^w}, \boxplus)$. [11]

– Відкидання операції $\boxplus$ веде до вразливості шифру до атаки мультиплікативних диференціалів. [3]

– Відкидання операції $\odot$ до компрометації шифру за допомогою криптоаналізу найменшого біта кожного слова. [1]

Також треба зауважити, що кожна з цих операцій має різні

властивості щодо розповсюдження змін:

– $\oplus$ - це лінійна операція, яка діє побітово, тому має найслабкішу форму розповсюдження;

– $\boxplus$ - нелінійна операція, яка має проштовхує зміни відповідно до вектору бітів переносу;

– $\odot$ - нелінійна операція, яка має найкращу форму розповсюдження, але може мати спеціальні класи значень, для яких її можна передбачити.

Зауваження. В деяких шифрах, які розглядаються в даній роботі (наприклад, в IDEA*), використовуються також обернені операції до вищезазначених. Це операції модульного віднімання $\boxminus$ (обернена до $\boxplus$) та ділення за модулем $\boxdot$ (обернена до $\odot$).

З огляду на запропоновані для використання операції, необхідно зазначити, що довжина підслів, на яке розбивається вхідне слово, в шифрах PES, IDEA обмежена 16 бітами. Причина криється в операції $\odot$: числа $2^{32}+1, 2^{64}+1, 2^{128}+1$ не є простими:

$$2^{32}+1 = 641 \cdot 6700417,$$

$$2^{64}+1 = 274177 \cdot 67280421310721,$$

$$2^{128}+1 = 59649589127497217 \cdot 5704689200685129054721,$$

в той час як $2^{16}+1$ - просте. Таким чином, деякі елементи для модульного множення просто не матимуть обернених, в кільцях зазначених модулів, що унеможливлює використання даної операції.

В шифрах родини MESH збільшено розмір раундового перетворення, але розмір підслів все одно є 16 бітів, як в шифрі IDEA.

Довжини підслів $W = \{2, 4, 8\}$ також можливі для використання. Шифр IDEA з такими довжинами підслів має назву mini-IDEA [11] та зазвичай використовується для перевірок гіпотез, які було б складно реалізувати на практиці для оригінального шифру.

1.4 Диференціальний аналіз шифрів на основі схеми Лая-Мессі

Диференціальний криптоаналіз - це техніка, винайдена Е. Біхамом та А. Шаміром в [2]. Початково вона була застосована до аналізу шифру DES зі зменшеною кількістю раундів, а згодом і до повного шифру DES. Ця техніка стала базовою, і була адаптована і до інших криптографічних примітивів, в тому числі і до шифрів на основі схеми Лая-Мессі. Стійкість до диференціального криптоаналізу тепер є вимогою для будь-якого нового криптографічного дизайну. [4]

Центральним поняттям диференціального криптоаналізу є *різниця*.

Означення 1.2. Різниця між двома бітовими векторами X та X^* - це вектор

$$\Delta X = X \bullet (X^*)^{-1},$$

де $\bullet$ - операція над групою, $(X^*)^{-1}$ - елемент, обернений до X^* відносно цієї операції.

Криптоаналітик може використовувати різниці пар вхідних і вихідних текстів, самостійно обираючи дані тексти. Спостерігаючи за еволюцією цих різниць, він може віднайти певні закономірності в їхніх розподілах та використати ці залежності для побудови атаки на криптографічний примітив. Таким чином, можливість передбачати різниці робить диференціальний криптоаналіз одним з найпотужніших з існуючих інструментів криптоаналітика.

Для визначення ймовірності отримання певних різниць існує поняття *диференціала* та його ймовірності.

Означення 1.3. Диференціал функції f над операціями $(\bullet, \circ)$ - це пара двійкових векторів (a, b), для яких існує значення x таке, що

виконується співвідношення:

$$f(x \circ a) \bullet (f(x))^{-1} = b,$$

де z^{-1} - елемент, обернений до z відносно операції $\bullet$.

Вектор a називають вхідною різницею, вектор b - вихідною різницею, $\circ$ та $\bullet$ - операціями різності на вході та виході відповідно.

Також диференціал іноді позначають як $a \underset{f}{\to} b$ або $a \to b$, якщо операція зрозуміла з контексту.

Означення 1.4. Ймовірністю диференціала (або диференціальною ймовірністю) ми будемо називати наступну величину:

$$DP^{f}_{\circ,\bullet}(a,b) = \overline{\sum_{x}} [f(x \circ a) \bullet (f(x))^{-1} = b].$$

Історично розвиток диференціального аналізу почався з дослідження випадку $\bullet \equiv \circ \equiv \oplus$. В даній роботі диференціальні характеристики будуть шукатись для певних алгебраїчних операцій, які будуть виступати в якості функції f. В центрі уваги будуть наступні диференціали:

1) $DP^{\boxplus}_{\oplus,\oplus}(\alpha, 0 \to \alpha) \equiv DP^{\boxplus}_{\oplus}(\alpha, 0 \to \alpha)$;

2) $DP^{\odot}_{\oplus,\oplus}(\alpha, 0 \to \alpha) \equiv DP^{\odot}_{\oplus}(\alpha, 0 \to \alpha)$;

3) $DP^{\otimes}_{\oplus,\oplus}(\alpha, 0 \to \alpha) \equiv DP^{\otimes}_{\oplus}(\alpha, 0 \to \alpha)$.

Варто зазначити, що для шифрів з великим простором вхідних слів (наприклад, для шифру IDEA з вхідним словом в 64 біти величина даного простору складає 2^{64} елементів) обрахунок всіх можливих диференціалів з метою визначення найбільш ймовірного з них є громіздкою обчислювальною задачею.

Ця проблема була розв'язана Х. Ліпмаа та Ш. Моріаі в роботі [9] для $DP^{\boxplus}_{\oplus}$. Ними було запропоновано ефективний обчислювальний алгоритм, який дозволяє будувати диференціали такого типу за $\Theta(log\ n)$, порівняно з методом безпосереднього обчислення за $\Omega(2^{2n})$.

Для $DP^{\odot}_{\oplus}$, на жаль, наразі невідомого алгоритму обчислення з

такою ефективністю, тому вони зазвичай обчислюються безпосередньо перебором. Також можливе проведення додаткового аналізу структури криптопримітивів, з метою визначення диференціалів особливого вигляду, які матимуть найбільшу ймовірність, отже, на яких є сенс зосереджуватись в першу чергу.

Висновки до розділу 1

Отже, було розглянуто порівняння схеми Лая-Мессі з більш популярною схемою Фейстеля, наведено приклади шифрів на її основі, згадано структурну слабкість схеми Лая-Мессі та описано існуючі модифікації для її посилення, а саме ортоморфні відображення та комбінації алгебраїчних операцій.

Оскільки другий спосіб є значно більш поширеним, то в наступних розділах ми зосередимо увагу саме на ньому, пропонуючи теоретичну атаку на описану схему та додатковий спосіб поєднання описаних операцій.

2 ДИФЕРЕНЦІАЛЬНА АТАКА НА ШИФР IDEA НА ОСНОВІ ВЛАСТИВОСТЕЙ ЙОГО КЛЮЧОВОГО СУМАТОРА

2.1 Опис структури шифру IDEA

Одним з найвідоміших шифрів на основі схеми Лая-Мессі є шифр IDEA, запропонований Х. Лаєм, Д. Мессі та Ш. Мерфі в 1991 році. [14] Це слово-орієнтований ітеративний блоковий шифр, який оперує над блоками довжини 64 біти, які розбиваються на підслова довжини 16 бітів кожне.

Нехай $X^{(i)} = (X_1^{(i)}, X_2^{(i)}, X_3^{(i)}, X_4^{(i)})$ - вхідне слово для i-го раунду шифрування, $X_j^{(i)} \in \mathbb{Z}_2^{16}, 1 \leqslant j \leqslant 4$. В таких позначеннях, $X^{(1)}$ - вхідний текст.

Один раунд шифрування можна розділити на 3 етапи: суматор з ключем, блок множення-додавання, вихідний блок.

Суматор з ключем виглядає наступним чином:

$$Y^{(i)} = (X_1^{(i)} \odot Z_1^{(i)}, X_2^{(i)} \boxplus Z_2^{(i)}, X_3^{(i)} \boxplus Z_3^{(i)}, X_4^{(i)} \odot Z_4^{(i)}),$$

після чого отримується впорядкована пара $(n_i, q_i) = (Y_1^{(i)} \oplus Y_3^{(i)}, Y_2^{(i)} \oplus Y_4^{(i)})$, яка подається на вхід блоку множення-додавання.

Результатом роботи цього блоку є впорядкована пара (r_i, s_i):

$$s_i = ((Z_5^{(i)} \odot n_i) \boxplus q_i) \odot Z_6^{(i)},$$

$$r_i = s_i \boxplus (Z_5^{(i)} \odot n_i).$$

Вихідний блок замішує результат роботи блоку множення-додавання з його входом, і переставляє слова наступним чином:

$X^{(i+1)} = \sigma(Y_1^{(i)} \oplus s_i, Y_2^{(i)} \oplus r_i, Y_3^{(i)} \oplus s_i, Y_4^{(i)} \oplus r_i) = (Y_1^{(i)} \oplus s_i, Y_3^{(i)} \oplus s_i, Y_2^{(i)} \oplus r_i, Y_4^{(i)} \oplus r_i)$.

$X^{(i+1)}$ є вхідним блоком для наступного раунду шифрування.

Процедура повторюється 8 раундів.

Після відпрацювання 8 раундів фінальне перетворення повторює перестановку σ (фактично, відміняючи її):

$$X^{(9)} = (Y_1^{(8)} \oplus s_8, Y_2^{(8)} \oplus r_8, Y_3^{(8)} \oplus s_8, Y_4^{(8)} \oplus r_8).$$

Фінальний шифротекст:

$$C = (X_1^{(9)} \odot Z_1^{(9)}, X_2^{(9)} \boxplus Z_2^{(9)}, X_3^{(9)} \boxplus Z_3^{(9)}, X_4^{(9)} \odot Z_4^{(9)}).$$

Алгоритм генерації ключів виглядає наступним чином:

1) Початковий ключ K зберігається в регістрі довжини 128 бітів і розбивається на 8 16-бітних підключів $Z_1^{(1)}$, $Z_1^{(2)}$, $Z_1^{(3)}$, $Z_1^{(4)}$, $Z_1^{(5)}$, $Z_1^{(6)}$, $Z_2^{(1)}$, $Z_2^{(2)}$.

2) Регістр зсувається вліво циклічно на 25 бітів і розбивається на 8 16-бітних підключів аналогічним чином.

3) Операція повторюється, поки не буде згенеровано необхідні 52 підключа.

Алгоритм дешифрування є дзеркальним відображенням алгоритму шифрування, зі зміною порядку раундових ключів та взяття обернених операцій $\boxdot, \boxminus$ до $\odot, \boxplus$ відповідно (рисунок 2.1).

Детальний криптоаналіз даного шифру, зокрема диференціальний, було проведено в роботі [15], зокрема, показано, що даний шифр має клас слабких ключів. Подальша атака в роботі [15] будувалася саме за припущення використання слабких ключів в процесі шифрування.

2.2 Опис нової диференціальної атаки на шифр IDEA

Можна помітити, що в структурі блоків додавання-множення шифру присутні 2 інваріанти, які є незалежними від значень раундових ключів:

$$n_i = Y_1^{(i)} \oplus Y_3^{(i)} = X_1^{(i+1)} \oplus X_2^{(i+1)},$$

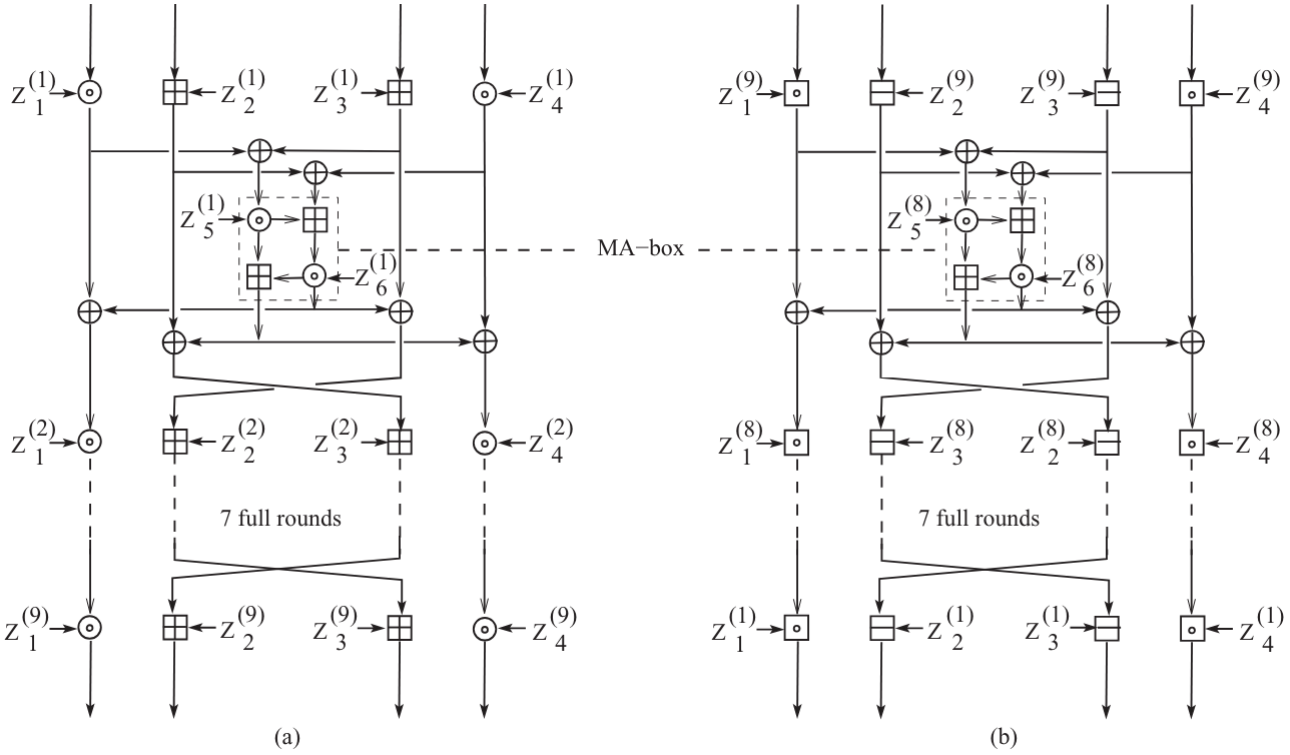

Рисунок 2.1 – Блок-схема роботи шифру IDEA: (a) шифрування, (b) розшифрування

$$q_i = Y_2^{(i)} \oplus Y_4^{(i)} = X_3^{(i+1)} \oplus X_4^{(i+1)}.$$

Ця структурна особливість дозволяє блоку додавання-множення зберегти інволютивність для простоти шифрування-дешифрування, але вона може бути використана при криптоаналізі шифру. Щоб завадити розповсюдженню цієї властивості на всі раунди, і існує блок суматора з ключем: $X_1^{(i)} \oplus X_3^{(i)} \neq Y_1^{(i)} \oplus Y_3^{(i)}$.

Розглянемо два повідомлення, різниця яких за побітовим додаванням має форму (α,α,α,α). Зі структури раундового перетворення видно, що якщо описана різниця пройде через ключовий суматор без змін, то вона збережеться впродовж всього раунду шифрування. Таким чином, можна розглядати лише ймовірність збереження різниці при проходженні через ключовий суматор без змін:

$$\mathbb{P} = (Pr_{x,y}((x \oplus \alpha) \boxplus y = (x \boxplus y) \oplus \alpha))^2 \cdot (Pr_{x,y}((x \oplus \alpha) \odot y = (x \odot y) \oplus \alpha))^2;$$

$$\mathbb{P} = (DP^{\boxplus}(\alpha, 0 \to \alpha))^2 \cdot (DP^{\odot}(\alpha, 0 \to \alpha))^2 = \mathbb{P}_1 \cdot \mathbb{P}_2.$$

Ймовірність $\mathbb{P}$ характеризує складність диференціальної атаки розпізнавання на шифр IDEA.

Такі диференціали було побудовано, найбільш ймовірні з них наведено в таблиці 2.1.

Таблиця 2.1 – Таблиця ймовірностей найбільш ймовірних диференціалів диф. атаки на шифр IDEA

α_{hex}	$DP^{\boxplus}_{\oplus}$	$DP^{\odot}_{\oplus}$	$\mathbb{P}$	$log_2(\mathbb{P})$
8080	0.5	0.000045	5.0625E-10	-30.87943095
4040	0.25	0.000045	1.26563E-10	-32.87943095
2020	0.25	0.000044	1.21E-10	-32.9442739
1010	0.25	0.000042	1.1025E-10	-33.07850229
0808	0.25	0.000039	9.50625E-11	-33.2923327
0404	0.25	0.000032	6.4E-11	-33.86313714
8001	0.5	0.000025	1.5625E-10	-32.57542476
8008	0.5	0.000025	1.5625E-10	-32.57542476
8010	0.5	0.000025	1.5625E-10	-32.57542476
8020	0.5	0.000025	1.5625E-10	-32.57542476
8040	0.5	0.000025	1.5625E-10	-32.57542476
8100	0.5	0.000025	1.5625E-10	-32.57542476
8200	0.5	0.000025	1.5625E-10	-32.57542476
8400	0.5	0.000025	1.5625E-10	-32.57542476
8800	0.5	0.000025	1.5625E-10	-32.57542476
9000	0.5	0.000025	1.5625E-10	-32.57542476
A000	0.5	0.000025	1.5625E-10	-32.57542476
C000	0.5	0.000025	1.5625E-10	-32.57542476
8004	0.5	0.000024	1.44E-10	-32.69321214
0202	0.25	0.000023	3.30625E-11	-34.81601323
8002	0.5	0.000022	1.21E-10	-32.9442739
8000	1	0.000015	2.25E-10	-32.04935595

Відповідно до таблиці 2.1, описана диференціальна атака може бути використана для побудови розпізнавача для одного раунду шифрування шифру IDEA з ймовірністю $\approx 2^{-31}$.

Даний результат показує, що ця атака носить теоретичний характер, оскільки сучасний розвиток обчислювальної техніки дозволяє за ефективний час провести атаку розпізнавання максимум на 2 раунди, в той час як IDEA проводить шифрування у 8 раундів.

З іншого боку, дана атака може проводитись незалежно від алгоритму розгортання раундових ключів, і взагалі не залежить від них.

2.3 Проведення запропонованої атаки на шифри PES, MESH-64, MESH-96, MESH-128

Подібна атака працюватиме за таким же принципом не тільки для шифру IDEA, а і для шифрів PES (попередник IDEA), MESH-64, MESH-96, MESH-128, оскільки вони мають подібну структуру і так само забезпечують стійкість схеми Лая-Мессі за допомогою ключового суматора. [10]

Блок-схеми шифрування та дешифрування цих шифрів наведені в рисунках Б.1, Б.2, Б.3, Б.4, Б.5, Б.6.

Атака на шифри PES та MESH-64 проводиться абсолютно аналогічно до атаки на шифр IDEA. Усі оцінки стійкості на диференціальні характеристики також будуть однаковими з огляду на однакові структури ключових суматорів та операцій, що в них задіяні.

В шифрах MESH-96 та MESH-128 використовуються, відповідно, 3 та 4 паралельні схеми Лая-Мессі, на відміну від 2 схем у PES, IDEA, MESH-64. Але операції, що використовуються в ключових суматорах, залишились ті ж самі, як і спосіб об'єднання результатів їх роботи.

З схем Б.3, Б.5 за аналогічних міркувань можна отримати побудувати наступні ймовірності диференціальної атаки на один раунд:

$$\mathbb{P}_{MESH96} = (DP^{\boxplus}(\alpha, 0 \to \alpha))^3 \cdot (DP^{\odot}(\alpha, 0 \to \alpha))^3 = (\mathbb{P}_1)^{\frac{3}{2}} \cdot (\mathbb{P}_2)^{\frac{3}{2}};$$

$$\mathbb{P}_{MESH128} = (DP^{\boxplus}(\alpha, 0 \to \alpha))^4 \cdot (DP^{\odot}(\alpha, 0 \to \alpha))^4 = (\mathbb{P}_1)^2 \cdot (\mathbb{P}_2)^2.$$

Таким чином, отримані результати аналізу стійкості шифру IDEA можуть бути прямо застосовані для обчислення стійкості шифрів MESH-96 та MESH-128.

Відповідно до таблиці 2.1, описана диференціальна атака може бути використана для побудови розпізнавача для одного раунду шифрування шифрів MESH-64, MESH-96, MESH-128 з ймовірностями $\approx 2^{-31}$, $2^{-46.5}$, 2^{-64} відповідно. Треба зазначити, що для шифру MESH-128 складність даної атаки майже збігається зі складністю повного перебору.

Висновки до розділу 2

В цьому розділі ми навели нову диференціальну атаку на шифр IDEA, яка носить теоретично-демонстративний характер, та побудували оцінки стійкості шифру від неї. Також показано застосування цієї атаки до сімейства шифрів MESH, і зв'язок між оцінками стійкості шифру IDEA та цих шифрів.

Результати показали, що описана диференціальна атака може бути використана для побудови розпізнавача для одного раунду шифрування шифрів IDEA, PES, MESH-64 з ймовірністю $\approx 2^{-31}$, і, відповідно, з ймовірністями $\approx 2^{-46.5}$ для шифру MESH-96 та $\approx 2^{-62}$ для шифру MESH-128.

Далі ми спробуємо навести модифікації існуючих ключових суматорів з метою покращення стійкості шифру IDEA, та вивести оцінки їх диференціальних характеристик.

3 ЗМІНА В ДИЗАЙНІ КЛЮЧОВОГО СУМАТОРА

Отже, ключовий суматор може бути слабким місцем в деяких шифрах на основі схеми Лая-Мессі. Однією з провідних ідей для його удосконалення є заміна операцій, наприклад, як в шифрі IDEA*.

3.1 Опис схеми шифру IDEA*

Даний блоковий шифр має таку саму структуру, як і оригінальний шифр IDEA, основна відмінність полягає в перестановці операцій [8].

Нехай $X^{(i)} = (X_1^{(i)}, X_2^{(i)}, X_3^{(i)}, X_4^{(i)})$ - повідомлення довжини 64 біти, розділене на 4 блоки по 16 бітів, де $X_j^{(i)} \in \mathbb{Z}_2^{16}, 1 \leqslant j \leqslant 4, 1 \leqslant i \leqslant 7$. В таких позначеннях $X^{(1)}$ - вхідне повідомлення.

Один раунд шифрування так само ділиться на 3 етапи, які мають трохи інший вигляд. Перший крок - ключовий суматор:

$$Y^{(i)} = (X_1^{(i)} \oplus Z_1^{(i)}, X_2^{(i)} \boxplus Z_2^{(i)}, X_3^{(i)} \boxplus Z_3^{(i)}, X_4^{(i)} \oplus Z_4^{(i)});$$

$$(n_i, q_i) = (Y_1^{(i)} \boxdot Y_3^{(i)}, Y_2^{(i)} \boxdot Y_4^{(i)}).$$

З ним йде АХ-блок (блок додавання та побітового додання), на відміну від АМ-блоку (додавання-множення) в IDEA. Результатом його роботи є впорядкована пара (r_i, s_i), де:

$$s_i = ((n_i \oplus Z_5^{(i)}) \boxplus q_i) \oplus Z_6^{(i)},$$

$$r_i = (n_i \oplus Z_5^{(i)}) \boxplus s_i.$$

Вихідний блок замішує результат роботи АХ-блоку з його входом, і переставляє підслова:

$$X^{(i+1)} = (Y_1^{(i)} \odot s_i, Y_3^{(i)} \odot s_i, Y_2^{(i)} \odot r_i, Y_4^{(i)} \odot r_i).$$

Вищеописані операції повторюються 7 раундів. З отриманого тексту $X^{(7)} = (X_1^{(7)}, X_2^{(7)}, X_3^{(7)}, X_4^{(7)})$ шифротекст отримується наступним чином:

$$C = (X_1^{(7)} \oplus Z_1^{(7)}, X_3^{(7)} \boxplus Z_2^{(7)}, X_2^{(7)} \boxplus Z_3^{(7)}, X_4^{(7)} \oplus Z_4^{(7)}).$$

При розшифруванні:

$$Y^{(i)} = (X_1^{(i)} \oplus Z_1^{(i)}, X_2^{(i)} \boxminus Z_2^{(i)}, X_3^{(i)} \boxminus Z_3^{(i)}, X_4^{(i)} \oplus Z_4^{(i)});$$

Результат розшифрування i-го раунду має вигляд:

$$(Y_1^{(i)} \boxdot s_i, Y_3^{(i)} \boxdot s_i, Y_2^{(i)} \boxdot r_i, Y_4^{(i)} \boxdot r_i).$$

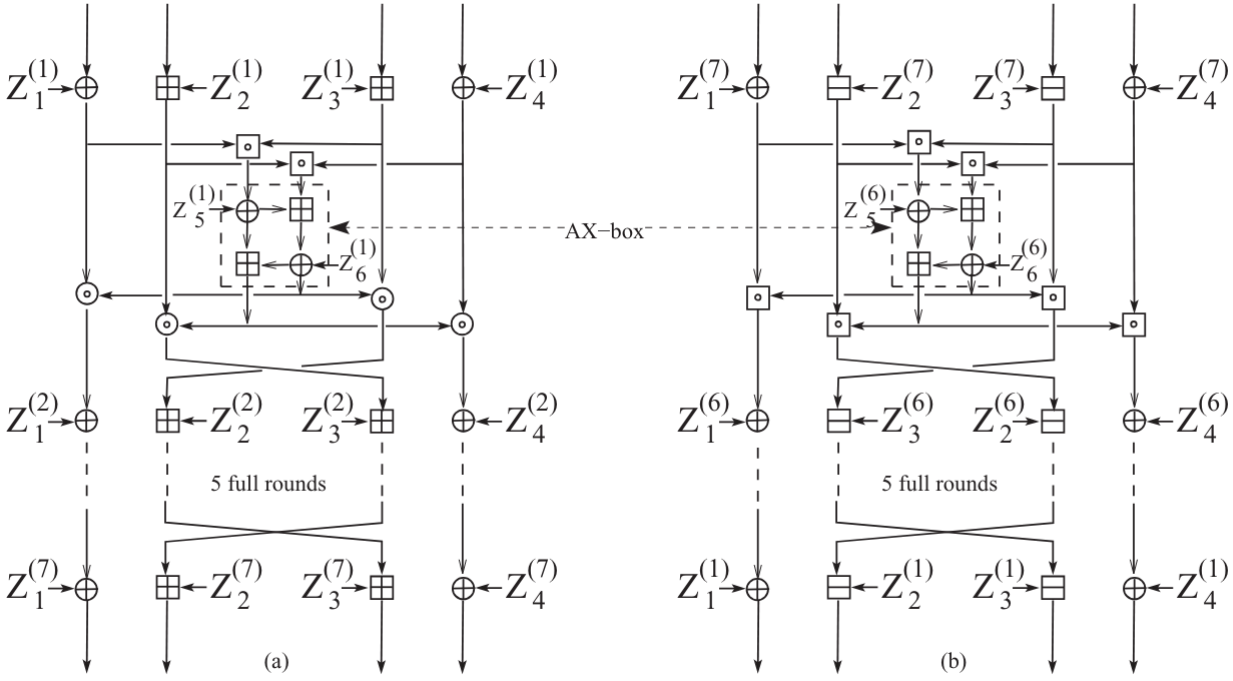

Рисунок 3.1 – Блок-схема роботи шифру IDEA*: (a) шифрування, (b) розшифрування

Схема розгортання раундових ключів є аналогічною до схеми розгортання ключів шифру MESH-64 [11].

3.2 Модифікація запропонованої атаки для шифру IDEA*

Можна помітити, що диференціальна атака, запропонована в розділі 2, незастосовна до шифру IDEA* через зміни в структурі його ключового суматора, а саме через спосіб поєднання його вихідних блоків за допомогою операції $\boxdot$. Додатково ця зміна ліквідовує наявність класу слабких ключів, характерних для шифру IDEA. Але згадану атаку можна модифікувати.

Розглянемо вектор вхідних різниць $(\alpha, \alpha \odot 2^w, \alpha \odot 2^w, \alpha)$ за операцією $\odot$ (для шифру IDEA різниці розглядались за операцією $\oplus$). Можна помітити, що якщо така різниця збережеться після проходження через ключовий суматор, то вона пройде в наступному вигляді через увесь раунд шифрування: $(\alpha \odot 2^w, \alpha, \alpha, \alpha \odot 2^w)$. Відповідно, на наступному раунді шифрування, за такої самої умови проходження, отримається початкова різниця.

Отже, необхідно розглядати ймовірність проходження таких різниць без змін через ключовий суматор. За міркуваннями, аналогічними до міркувань з атаки на оригінальний шифр IDEA, така ймовірність буде дорівнювати:

$$\mathbb{P}^* = (DP^{\oplus}_{\odot}(\alpha, 1 \to \alpha \odot 2^w))^2 \cdot (DP^{\boxplus}_{\odot}(\alpha \odot 2^w, 1 \to \alpha))^2 =$$

$$= (Pr_{x,y}((x \odot \alpha) \oplus y = (x \oplus y) \odot \alpha \odot 2^w))^2 \cdot (Pr_{x,y}((x \odot \alpha \odot 2^w) \boxplus y = (x \boxplus y) \odot \alpha))^2.$$

Зауваження. Оскільки нейтральним елементом за операцією $\odot$ є елемент 1, то він і бере участь в диференціалі, з огляду на те, щоб різниця по ключах була нульовою, оскільки на них криптоаналітик впливати не може.

Найбільші диференціали для деяких вхідних різниць α наведено в таблиці 3.1. Як видно, більша частина диференціалів близька за значенням до 2^{-64}, що відповідає повному перебору ключів, тому ефективність даної атаки навіть на один раунд є сумнівною.

Таблиця 3.1 – Найбільші диференціали характеристики для шифру IDEA*

α_{hex}	$DP^{\oplus}_{\odot}$	$DP^{\boxplus}_{\odot}$	P^*	$log_2(P^*)$
0100	0.001674	0.000023	1.4824E-15	-49.26098274
0010	0.000104	0.000015	2.4336E-18	-58.51161365
1000	0.000104	0.000015	2.4336E-18	-58.51161365
0004	0.000026	0.000015	1.521E-19	-62.51161365
0040	0.000026	0.000015	1.521E-19	-62.51161365
0400	0.000026	0.000015	1.521E-19	-62.51161365
4000	0.000026	0.000015	1.521E-19	-62.51161365
0008	0.000018	0.000015	7.29E-20	-63.57264308
2000	0.000018	0.000015	7.29E-20	-63.57264308
0020	0.000017	0.000015	6.5025E-20	-63.7375674
0800	0.000017	0.000015	6.5025E-20	-63.7375674
8004	0.000016	0.000015	5.76E-20	-63.91249309
8020	0.000016	0.000015	5.76E-20	-63.91249309
0404	0.000016	0.000015	5.76E-20	-63.91249309
2020	0.000016	0.000015	5.76E-20	-63.91249309
0001	0.000015	0.000015	5.0625E-20	-64.09871189
0002	0.000015	0.000015	5.0625E-20	-64.09871189

3.3 Введення нової операції

Оскільки підхід до заміни та перестановки порядку виконання операцій в шифрі IDEA показав свою ефективність, спробуємо розширити набір операцій, які використовуються в ключовому суматорі. Розглянемо наступну операцію:

$$f(x,y) = x \otimes y = (x \boxplus 1) \odot (y \boxplus 1) \boxminus 1$$

Зауваження. Оскільки $x \in [0, 2^n - 1], y \in [0, 2^n - 1]$, то $(x \boxplus 1) \odot (y \boxplus 1) \in [1, 2^n]$, відповідно $x \otimes y \in [0, 2^n - 1]$, отже ця операція, на відміну від модульного множення в оригінальному шифрі IDEA, виконується для природного представлення чисел двійковими векторами

без додаткових уточнень. З іншого боку, введена операція є складнішою з точки зору обчислення, оскільки вона є композицією декількох операцій одночасно, але таке ускладнення може бути застосоване для збільшення стійкості шифру.

Диференціали за операцією $\oplus$ мають декілька особливих властивостей. Сформулюємо їх у вигляді лем 3.1 та 3.2:

Лема 3.1. $DP^{\otimes}_{\oplus}((2^n-1),(2^n-1)\to 0)=1$

Доведення. За означенням:

$$DP^{\otimes}_{\oplus}(2^n-1,2^n-1\to 0)=\overline{\sum_{x,y}}\,[f(x\oplus(2^n-1),y\oplus(2^n-1))\oplus f(x,y)=0]=$$

$$=\overline{\sum_{x,y}}\,[(((x\oplus(2^n-1))+1)((y\oplus(2^n-1))+1)-1)\oplus((x+1)(y+1)-1)=0]=\mathbb{T}.$$

Зауважимо, що $x\oplus(2^n-1)=2^n-1-x$, оскільки це є інвертацією всіх бітів числа. Також зауважимо, що $(2^n-x)\pmod{2^n}=(2^n-x)\pmod{2^n+1}$, оскільки $x\in[0,2^n-1]$, відповідно:

$$\mathbb{T}=\overline{\sum_{x,y}}\,[((2^n-x)(2^n-y)-1)\oplus((x+1)(y+1)-1)=0]=$$

$$=\overline{\sum_{x,y}}\,[2^{2n}-x\odot 2^n-y\odot 2^n+xy-1)\oplus((x+1)(y+1)-1)=0].$$

Очевидно, що:

$2^{2n}\equiv(-1)^2\pmod{2^n+1}\equiv 1\pmod{2^n+1}$;

$-x\cdot 2^n\equiv x\pmod{2^n+1}$;

$-y\cdot 2^n\equiv y\pmod{2^n+1}$.

Отже:

$$\mathbb{T}=\overline{\sum_{x,y}}\,[(x+y+xy)\oplus(xy+x+y)=0]=1.$$

□

Наведемо ще один диференціал особливого вигляду:

Лема 3.2. $DP^{\otimes}_{\oplus}(0, 2^n - 1 \to 2^n - 1) = DP^{\otimes}_{\oplus}(2^n - 1, 0 \to 2^n - 1) = 1$

Доведення. В силу симетричності операції $\otimes$ можна розглядати лише один з диференціалів:

$$DP^{\otimes}_{\oplus}(0, 2^n \boxminus 1 \to 2^n \boxminus 1) = \overline{\sum_{x,y}} [f(x, y \oplus (2^n \boxminus 1)) \oplus f(x, y) = 2^n \boxminus 1] =$$

$$= \overline{\sum_{x,y}} [((x \boxplus 1) \odot (y \oplus (2^n \boxminus 1)) \boxplus 1) \boxminus 1) \oplus ((x \boxplus 1) \odot (y \boxplus 1) \boxminus 1) = 2^n \boxminus 1] = \mathbb{T}.$$

За аналогічними міркуваннями, наведеними в 3.1:

$(x \boxplus 1) \pmod{2^n} = (x \boxplus 1) \pmod{2^n \boxplus 1}$;

$(2^n \boxminus y \boxminus 1) \pmod{2^n} = (2^n \boxminus y \boxminus 1) \pmod{2^n \boxplus 1}$.

Отже:

$$\mathbb{T} = \overline{\sum_{x,y}} [((x \boxplus 1) \odot (2^n \boxminus y) \boxminus 1) \oplus (xy \boxplus x \boxplus y) = 2^n \boxminus 1] =$$

$$= \overline{\sum_{x,y}} [(2^n \odot x \boxplus 2^n \boxminus x \odot y \boxminus y \boxminus 1) \oplus (xy \boxplus x \boxplus y) = 2^n \boxminus 1] =$$

$$= \overline{\sum_{x,y}} [(2^n \boxminus 1 \boxminus (xy \boxplus x \boxplus y)) \oplus (xy \boxplus x \boxplus y) = 2^n \boxminus 1] =$$

$$= \overline{\sum_{x,y}} [(xy \boxplus x \boxplus y) \oplus (2^n \boxminus 1) \oplus (xy \boxplus x \boxplus y) = 2^n \boxminus 1] = 1$$

□

Отже, подібно до диференціалів за $\boxplus$, для яких особливою різницею є 2^{n-1}, запропонована операція також має особливі значення, які треба розглядати при диференціальному криптоаналізі, оскільки вони мають потенціал бути найбільшими і суттєво зменшити стійкість шифру.

3.4 Зміна в дизайні ключового суматора

Розглянемо два модифікованих ключових суматора шифру IDEA:

$$Y'^{(i)} = (X_1^{(i)} \otimes Z_1^{(i)}, X_2^{(i)} \boxplus Z_2^{(i)}, X_3^{(i)} \boxplus Z_3^{(i)}, X_4^{(i)} \otimes Z_4^{(i)}), \tag{3.1}$$

$$Y''^{(i)} = (X_1^{(i)} \odot Z_1^{(i)}, X_2^{(i)} \otimes Z_2^{(i)}, X_3^{(i)} \otimes Z_3^{(i)}, X_4^{(i)} \odot Z_4^{(i)}). \tag{3.2}$$

Для оригінального шифру IDEA та для шифрів з даними суматорами вже було знайдено різниці α, які визначають найбільші можливі значення імовірності $\mathbb{P}$. Одержані результати наведено у таблиці 2.1.

Проведемо таку саму атаку для шифрів з ключовими суматорами (3.1) та (3.2), і порівняємо отримані результати:

Можна побачити, що структурна зміна (3.1) призвела до незначного зменшення стійкості шифру, в той час як (3.2), навпаки, значно посилила її для атаки, описаної вище. Виграш в стійкості між (3.1) та (3.2) є в середньому майже квадратичним:

$$P(Y''^{(i)}) \approx (P(Y'^{(i)}))^{1.94}.$$

Виграш в стійкості між оригінальним суматором та (3.2) є меншим, але також суттєвим:

$$P(Y''^{(i)}) \approx (P(Y^{(i)}))^{1.81}.$$

Отже, для 8 раундів шифрування оригінальний шифр IDEA проводив по 16 операцій $\boxplus$ та $\odot$ в ключових суматорах. Для роботи ключового суматора (3.1) на 8 раундах необхідно 48 $\boxplus$, 16 $\odot$, 16 $\boxminus$. Для роботи ключового суматора (3.2), відповідно, 32 $\boxplus$, 32 $\odot$, 16 $\boxminus$.

Таким чином, покращення стійкості ключового суматора (3.2) досягається шляхом певного ускладнення обчислень, в той час як ключовий суматор (3.1) не дає покращення в стійкості навіть за рахунок ускладнення обрахунків.

Зауваження. Можна спостерігати, що ймовірність диференціалу

Таблиця 3.2 – Порівняльна таблиця ймовірностей найбільш ймовірних диференціалів диф. атаки на шифр IDEA

α_{hex}	$log_2(\mathbb{P}(Y^{(i)}))$	$log_2(\mathbb{P}(Y'^{(i)}))$	$log_2(\mathbb{P}(Y''^{(i)}))$	α_{hex}	$log_2(\mathbb{P}(Y^{(i)}))$	$log_2(\mathbb{P}(Y'^{(i)}))$	$log_2(\mathbb{P}(Y''^{(i)}))$
0001	-34.05	-31.95	-62	8010	-32.58	-31.15	-59.72
0002	-34.05	-31.95	-62	8020	-32.58	-31.15	-59.72
0004	-34.05	-31.95	-62	8040	-32.58	-31.15	-59.72
0008	-34.05	-31.95	-62	8080	-30.88	-30	-56.88
0010	-34.05	-31.95	-62	8100	-32.58	-31.15	-59.72
0020	-34.05	-31.95	-62	8200	-32.58	-31.15	-59.72
0040	-34.05	-31.95	-62	8400	-32.58	-31.15	-59.72
0080	-34.05	-31.95	-62	8800	-32.58	-31.15	-59.72
0100	-34.05	-31.95	-62	9000	-32.58	-31.15	-59.72
0200	-34.05	-31.95	-62	A000	-32.58	-31.15	-59.72
0400	-34.05	-31.95	-62	C000	-32.58	-31.15	-59.72
0800	-34.05	-31.95	-62	0101	-36.05	-32	-60.05
1000	-34.05	-31.95	-62	0202	-34.82	-32	-58.82
2000	-34.05	-31.95	-62	0404	-33.86	-32	-57.86
4000	-34.05	-31.95	-62	0808	-33.29	-32	-57.29
8000	-32.05	-29.95	-62	1010	-33.08	-32	-57.08
8001	-32.58	-31.15	-59.72	2020	-32.94	-32	-56.95
8002	-32.94	-31.15	-60.09	4040	-32.88	-32	-56.88
8004	-32.69	-31.15	-59.84	8080	-30.88	-30	-56.88
8008	-32.58	-31.15	-59.72	FFFF	-62	-29.95	-32.05

для $\alpha_{hex} = (FFFF)$ є оберненою до загальної картини між $Y^{(i)}$ та $Y''^{(i)}$.

Дана особливість є наслідком леми 3.2, оскільки диференціал за додаванням для такої різниці є максимальним. Це, безумовно, є слабкістю нововведеної операції.

В той же час, операція $\boxplus$ також має диференціали з ймовірністю один:

$$DP^{\boxplus}_{\oplus}\left((8000_{hex}, 8000_{hex}) \to 8000_{hex}\right) = 1,$$

тому наявність таких диференціалів не є підставою для відмови від використання операцій, що їх мають.

Висновки до розділу 3

Було запропоновано нову операцію, яка є поєднанням операцій, що вже використовувались в шифрах на основі схеми Лая-Мессі, з метою покращити стійкість шифру IDEA до запропонованої диференціальної атаки. Виведено стійкість шифру зі зміненими ключовими суматорами то проведено їх порівняння з стійкістю оригінального шифру.

Запропонована диференціальна атака хоч і носить теоретичний характер, але показує важливість надійного ключового суматора, слабкість якого може скомпрометувати увесь криптографічний фреймворк на основі схеми Лая-Мессі.

ВИСНОВКИ

В ході огляду літератури було розглянуто існуючі шифри на основі схеми Лая-Мессі, проведено аналіз переваг даних шифрів над шифрами, що базуються на схемі Фейстеля. Також визначено способи захисту схеми Лая-Мессі від її структурної слабкості, описано існуючі модифікації для її посилення, а саме ортоморфні відображення та комбінації алгебраїчних операцій.

Було запропоновано нову диференціальну атаку на шифр IDEA, яка носить теоретично-демонстративний характер, та побудовано оцінки стійкості шифрів PES, IDEA, MESH-64, MESH-96, MESH-128 від неї. Результати дослідження показали, що описана диференціальна атака може бути використана для побудови розпізнавача для одного раунду шифрування шифрів IDEA, PES, MESH-64 з ймовірністю $\approx 2^{-31}$, і, відповідно, з ймовірностями $\approx 2^{-46.5}$ для шифру MESH-96 та $\approx 2^{-62}$ для шифру MESH-128.

Також було запропоновано зміну в структурі ключових суматорів вищеописаних шифрів, суть якої полягає в поєднанні вже використовуваних алгебраїчних операцій, з метою покращити стійкість шифрів до наведеної диференціальної атаки. Отримана диференціальна характеристика стійкості згаданих шифрів зі зміненими ключовими суматорами та проведено її порівняння з оригінальною.

ПЕРЕЛІК ПОСИЛАНЬ

[1] Eli Biham, Orr Dunkelman та Nathan Keller. «New Cryptanalytic Results on IDEA». В: *Advances in Cryptology – ASIACRYPT 2006*. За ред. Xuejia Lai та Kefei Chen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, с. 412—427. ISBN: 978-3-540-49476-8.

[2] Eli Biham та Adi Shamir. «Differential Cryptanalysis of DES-like Cryptosystems». В: *J. Cryptology* 4 (1991), с. 3—72. DOI: 10.1007/BF00630563.

[3] Nikita Borisov та ін. «Multiplicative Differentials». В: *Fast Software Encryption*. За ред. Joan Daemen та Vincent Rijmen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, с. 17—33.

[4] Lawrence Brown та ін. «Improving resistance to differential cryptanalysis and the redesign of LOKI». В: *Advances in Cryptology — ASIACRYPT '91*. За ред. Hideki Imai, Ronald L. Rivest та Tsutomu Matsumoto. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, с. 36—50.

[5] Zong Duo Daia, Solomon W. Golomb та Guang Gong. «Generating linear ortomorphisms without repetition». Англ. В: *Discrete mathematics* 205.1 (1999), с. 47—55.

[6] Lorenzo Grassi. *On Generalizations of the Lai-Massey Scheme*. Cryptology ePrint Archive, Paper 2022/1245. https://eprint.iacr.org/2022/1245. 2022. URL: https://eprint.iacr.org/2022/1245.

[7] M. Hall та L. Paige. «Complete mappings of finite groups». Англ. В: *Pacific Journal of Mathematics* 5.4 (1955), с. 541—550.

[8] Liran Lerman, Jorge Nakahara та Nikita Veshchikov. «Improving block cipher design by rearranging internal operations». В: *2013 International Conference on Security and Cryptography (SECRYPT)*. 2013, с. 1—12.

[9] Helger Lipmaa та Shiho Moriai. *Efficient Algorithms for Computing Differential Properties of Addition*. Cryptology ePrint Archive, Paper 2001/001. 2001.

[10] Jorge Nakahara та ін. «The MESH Block Ciphers». В: *Information Security Applications*. За ред. Ki-Joon Chae та Moti Yung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, с. 458—473.

[11] Nakahara J.Jr. *Lai-Massey Cipher Designs. History, Design Criteria and Cryptanalysis*. Англ. 2018. URL: https://link.springer.com/book/10.1007/978-3-319-68273-0.

[12] Serge Vaudenay. «Decorrelation: A Theory for Block Cipher Security». В: *Journal of Cryptology* 16 (вер. 2003). DOI: 10.1007/s00145-003-0220-6.

[13] Serge Vaudenay. «On the Lai-Massey Scheme». В: *Advances in Cryptology - ASIACRYPT'99*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, с. 8—19.

[14] Lai Xuejia. «On the Design and Security of Block Ciphers». Дис. ... док. Swiss Federal Institute of Technology, 1992.

[15] Lai Xuejia, James Massey та Sean Murphy. «Markov Ciphers and Differential Cryptanalysis». В: *1991 Advances in Cryptology - EUROCRYPT '91*. 1991, с. 1—12.

ДОДАТОК А ТЕКСТИ ПРОГРАМ

А.1 Код програми для обрахунку диференціалів

```
1  #include <vector>
2  #include <string>
3  #include <vector>
4  #include <iostream>
5  #include <fstream>
6  #include <utility>
7  #include <fstream>
8  #include <stdexcept>
9  #include <sstream>
10 #include <iomanip>
11
12 using namespace std;
13
14 // Creates truth table for multiplication modulo (base + 1)
15 unsigned short** createMulTable(unsigned int base) {
16     unsigned int base_mul = base + 1;
17
18     unsigned short** table = new unsigned short* [base];
19     for (unsigned int i = 0; i < base; ++i)
20         table[i] = new unsigned short[base];
21
22     for (unsigned int i = 0; i < base; i++) {
23         //cout << "Stage x: " << i << "\n";
24         for (unsigned int j = 0; j < base; j++) {
25             table[i][j] = (i * j) % (base_mul);
26         }
27     }
28
29     return table;
30 }
31
32 // Creates truth table for operation X
33 unsigned short** createOperationTable(unsigned int base) {
34     unsigned int base_mul = base + 1;
35
```

```
36      unsigned short** table = new unsigned short* [base];
37      for (unsigned int i = 0; i < base; ++i)
38          table[i] = new unsigned short[base];
39
40      for (unsigned int i = 0; i < base; i++) {
41          //cout << "Stage x: " << i << "\n";
42          for (unsigned int j = 0; j < base; j++) {
43              table[i][j] = ((i + 1) * (j + 1) % (base_mul)) - 1;
44          }
45      }
46
47      return table;
48  }
49
50  // Returns one differential by xor for operation with truth table for specified
        difference
51  long double calc_tabled_diff(unsigned int base, unsigned short alpha, unsigned short
        ** table) {
52      cout << "Starting calculating tabled differential for alpha: " + to_string(alpha
            ) + "\n";
53      long double sum = 0;
54      for (unsigned int x = 0; x < base; x++) {
55          for (unsigned int y = 0; y < base; y++) {
56              unsigned short left = table[x ^ alpha][y];
57              unsigned short right = table[x][y] ^ alpha;
58              if (left == right) sum = sum + 1;
59          }
60      }
61      return sum / pow(base, 2);
62  }
63
64  // Returns one differential by xor for operation MODADD for specified difference
65  long double calc_add_diff(unsigned int base, unsigned short alpha) {
66      cout << "Starting calculating add differential for alpha: " + to_string(alpha) +
            "\n";
67      long double sum = 0;
68      for (unsigned long int x = 0; x < base; x++) {
69          for (unsigned long int y = 0; y < base; y++) {
70              unsigned short left = ((x ^ alpha) + y) % base;
```

```
71                  unsigned short right = ((x + y) % base) ^ alpha;
72                  if (left == right) sum = sum + 1;
73              }
74          }
75          return sum / pow(base, 2);
76      }
77
78      // Returns vector with all differentials by xor for operation with a truth table for
            all differences
79      vector<pair<unsigned short, long double>> get_tabled_diffs(unsigned int base,
            unsigned short** table) {
80          vector<pair<unsigned short, long double>> differentials_tabled;
81          for (unsigned short alpha = 0; alpha < base; alpha++) {
82              long double tabled_diff = calc_tabled_diff(base, alpha, table);
83              differentials_tabled.push_back(make_pair(alpha, tabled_diff));
84          }
85          return differentials_tabled;
86      }
87
88      // Returns vector with all differentials by xor for operation with a truth table for
            specified differences
89      vector<pair<unsigned short, long double>> get_tabled_diffs(unsigned int base,
            unsigned short** table, unsigned short* alphas, int size) {
90          vector<pair<unsigned short, long double>> differentials_tabled;
91          for (unsigned short i = 0; i < size; i++) {
92              long double tabled_diff = calc_tabled_diff(base, alphas[i], table);
93              differentials_tabled.push_back(make_pair(alphas[i], tabled_diff));
94          }
95          return differentials_tabled;
96      }
97
98      // Returns vector with all differentials by xor for operation MODADD for all
            possible differences
99      vector<pair<unsigned short, long double>> get_add_diffs(unsigned int base) {
100         vector<pair<unsigned short, long double>> differentials_add;
101         for (unsigned short alpha = 0; alpha < base; alpha++) {
102             long double add_diff = calc_add_diff(base, alpha);
103             differentials_add.push_back(make_pair(alpha, add_diff));
104         }
```

```
105     return differentials_add;
106 }
107
108 // Returns vector with all differentials by xor for operation MODADD for specified
        differences
109 vector<pair<unsigned short, long double>> get_add_diffs(unsigned int base, unsigned
        short* alphas, int size) {
110     vector<pair<unsigned short, long double>> differentials_add;
111     for (unsigned short i = 0; i < size; i++) {
112         long double add_diff = calc_add_diff(base, alphas[i]);
113         differentials_add.push_back(make_pair(alphas[i], add_diff));
114     }
115     return differentials_add;
116 }
117
118 int main() {
119     //unsigned int base = pow(2, 8);
120     unsigned int base = pow(2, 16);
121     long double square_base = pow(base, 2);
122     //unsigned short alphas_mul[32] = { 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024,
            2048, 4096, 8192, 16384, 32768, 32769, 32770, 32772, 32776, 32784, 32800,
            32832, 32896, 33024, 33280, 33792, 34816, 36864, 40960, 49152, 65533 };
123     unsigned short alphas_operation[40] = { 1, 2, 4, 8, 16, 32, 64, 128, 256, 512,
            1024, 2048, 4096, 8192, 16384, 32768, 32769, 32770, 32772, 32776, 32784,
            32800, 32832, 32896, 33024, 33280, 33792, 34816, 36864, 40960, 49152, 257,
            514, 1028, 2056, 4112, 8224, 16448, 32896, 65535 };
124     cout << "Starting calculating differentials...\n";
125
126     //auto table = createMulTable(base);
127     auto table = createOperationTable(base);
128
129     //vector<pair<unsigned short, long double>> differentials_tabled =
            get_tabled_diffs(base, table);
130     //vector<pair<unsigned short, long double>> differentials_add = get_add_diffs(
            base);
131     vector<pair<unsigned short, long double>> differentials_tabled =
            get_tabled_diffs(base, table, alphas_operation, 40);
132     vector<pair<unsigned short, long double>> differentials_add = get_add_diffs(base
            , alphas_operation, 40);
```

```
133
134     cout << "Writing to file...\n";
135
136     //ofstream ostream("MULTIPLICATION_differentials_mod_" + to_string(base) + ".csv
            ");
137     //ostream << "alpha, diff_mul, diff_add\n";
138     ofstream ostream("OPERATION_differentials_mod_" + to_string(base) + ".csv");
139     ostream << "alpha, diff_operation, diff_add\n";
140     for (unsigned int i = 0; i < differentials_tabled.size(); ++i) {
141         ostream << to_string(differentials_tabled[i].first) << ", " << to_string(
                differentials_tabled[i].second) << ", " << to_string(differentials_add[i
                ].second) << "\n";
142     }
143     ostream.close();
144     cout << "Finished writing to file.\n";
145         return 0;
146 }
```

ДОДАТОК Б ВЕЛИКІ РИСУНКИ ТА ТАБЛИЦІ

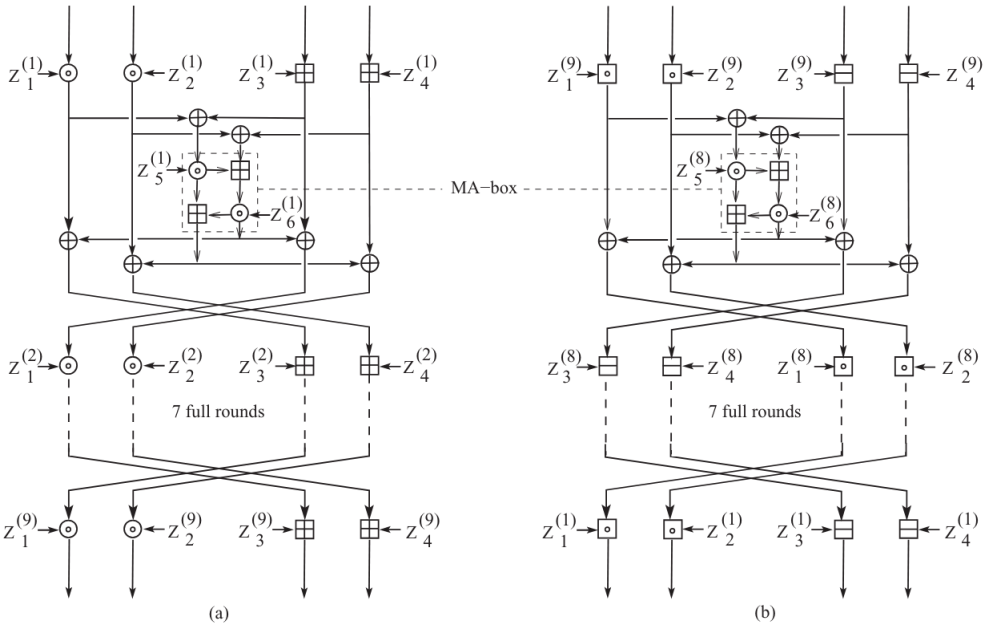

Рисунок Б.1 – Блок-схема шифру PES: (a) - шифр., (b) - розшифр.

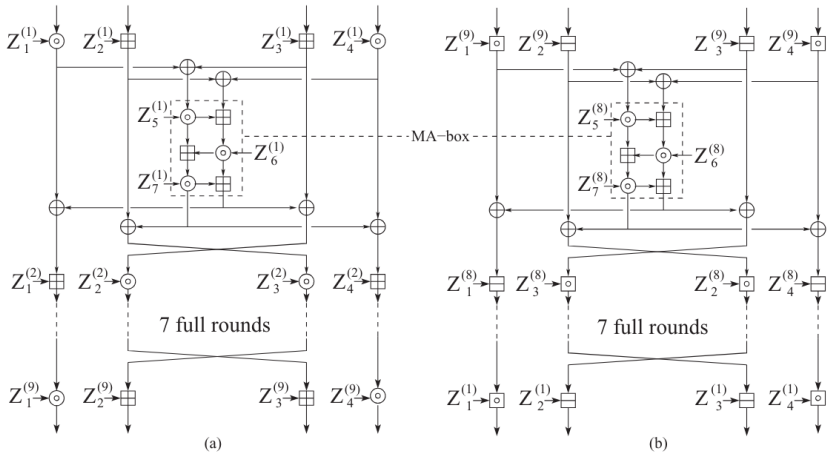

Рисунок Б.2 – Блок-схема шифру MESH-64: (a) - шифр., (b) - розшифр.

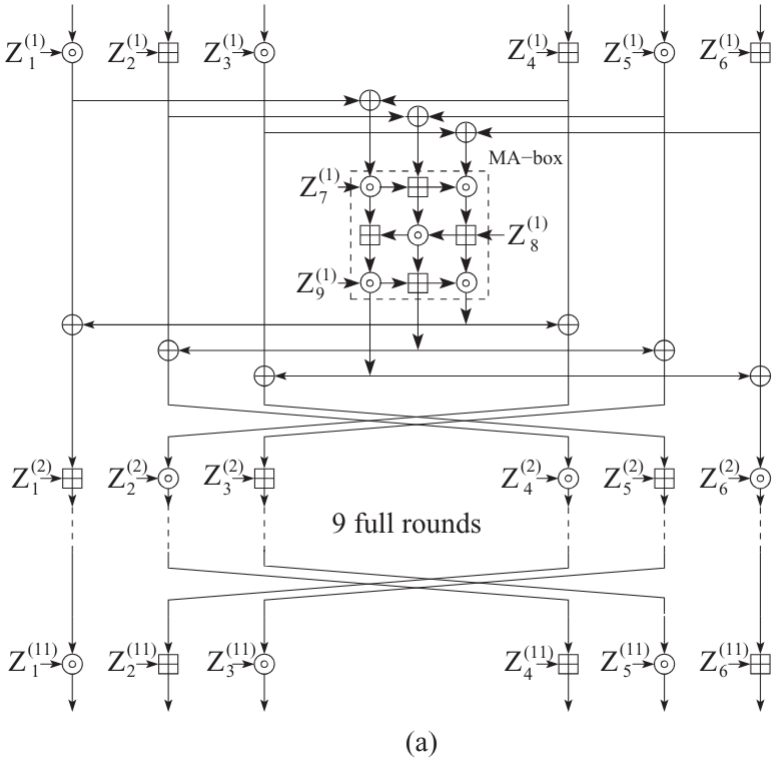

Рисунок Б.3 – Структурна схема шифру MESH-96: (a) - шифрування

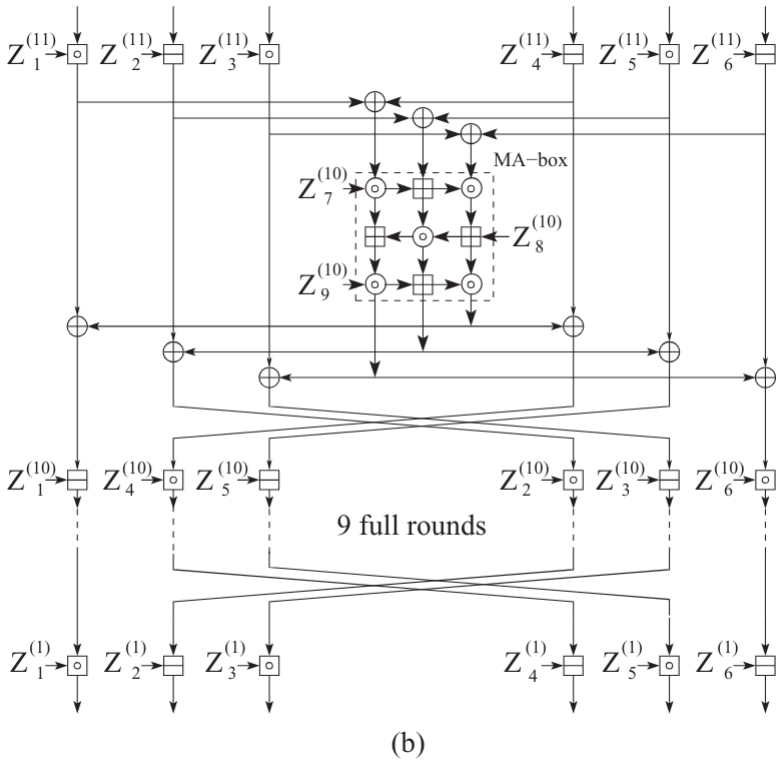

Рисунок Б.4 – Структурна схема шифру MESH-96: (b) - розшифрування

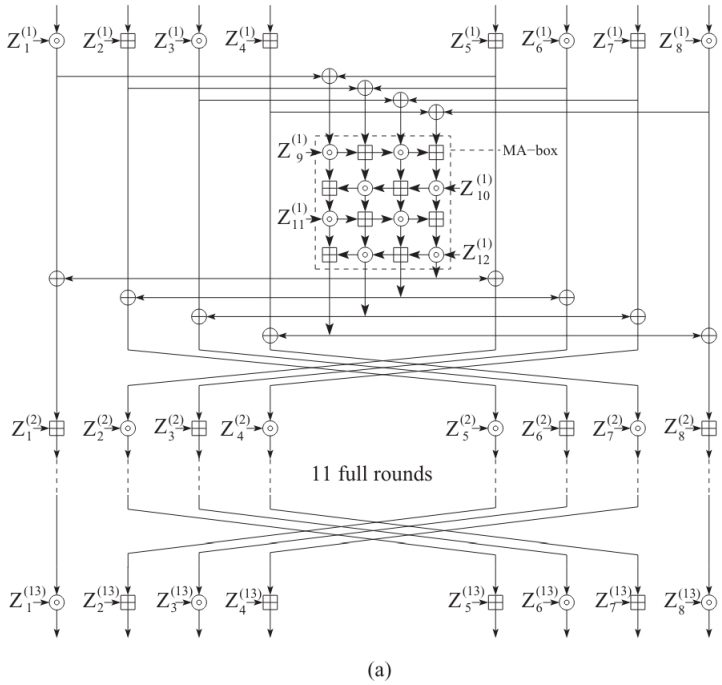

Рисунок Б.5 – Структурна схема шифру MESH-128: (a) - шифрування

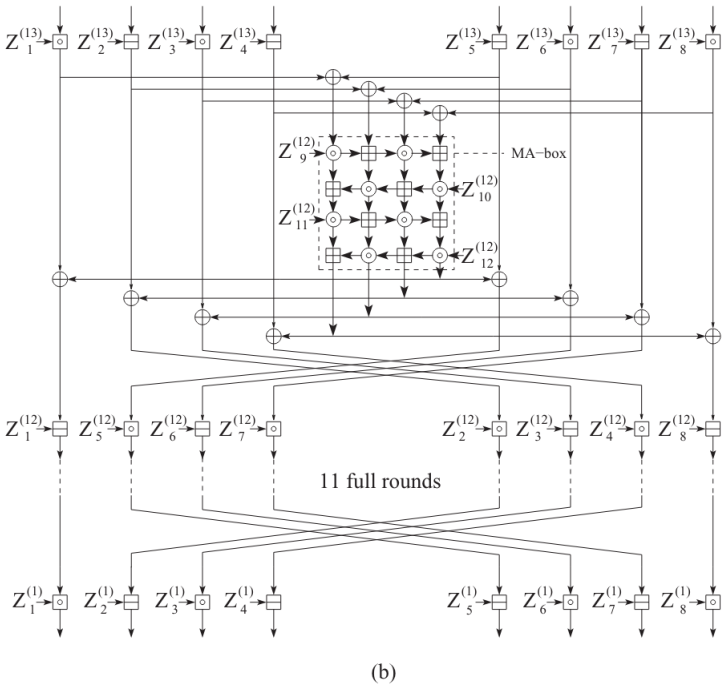

Рисунок Б.6 – Структурна схема шифру MESH-128: (b) - розшифрування