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

«На правах рукопису»
УДК 004.056.55:512.6+519.24

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

Дипломна робота

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

зі спеціальності: 113 Прикладна математика
на тему: **«Диференціальні атаки збоїв на шифр Qalqan»**

Виконав:
студент IV курсу, групи ФІ-03
Недождій Максим Андрійович _____

Керівник:
доцент кафедри ММЗІ, к.т.н.
Яковлєв Сергій Володимирович _____

Рецензент:
посада, степінь, звання
_____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

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 — перший (бакалаврський)
Спеціальність — 113 Прикладна математика,
ОПП «Математичні методи криптографічного захисту інформації»

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

ЗАВДАННЯ
на дипломну роботу

Студент: Недождій Максим Андрійович

1. Тема роботи: «*Диференціальні атаки збоїв на шифр Qalqan*», науковий керівник дипломної роботи: доцент кафедри 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: інформаційні процеси в системах захисту інформації

4. Предмет дослідження: стійкість симетричного блокового шифру Qalqan до атак на реалізацію

5. Перелік завдань:

1) провести огляд опублікованих джерел за тематикою дослідження;

2) проаналізувати структуру та особливості шифру Qalqan;

3) побудувати атаку збоїв на ключ останнього раунду шифру Qalqan та проаналізувати її ефективність;

4) побудувати атаку збоїв на ключі проміжних раундів шифру Qalqan та проаналізувати її ефективність.

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

7. Орієнтовний перелік публікацій: Результати були представлені на II Міжнародній науково-практичній конференції «Кіберборотьба: розвідка, захист та протидія» (23-24 квітня 2024 р., м. Київ, Україна) [12], XXII Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» (13-17 травня 2024 р., м. Київ, Україна) [13].

8. Дата видачі завдання: 10 вересня 2023 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	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	Аналіз структури та особливостей шифру Qalqan	Листопад-грудень 2023 р.	Виконано
4	Побудова атаки збоїв на ключ останнього раунду шифру Qalqan та аналіз її ефективності.	Січень-лютий 2024 р.	Виконано
5	Побудова атаки збоїв на ключ проміжних раундів шифру Qalqan та аналіз її ефективності.	Березень-квітень 2024 р.	Виконано
6	Оформлення дипломної роботи	Травень 2024 р.	Виконано

Студент _____ Максим НЕДОЖДІЙ

Керівник _____ Сергій ЯКОВЛЄВ

РЕФЕРАТ

Кваліфікаційна робота містить: 60 стор., 5 рисунків, 5 таблиць, 14 джерел.

У даній роботі досліджується застосування диференціальних атак збоїв (DFA) на модифікований шифр Qalqan, що є кандидатом у національний стандарт Республіки Казахстан. Описано шифр і його модифікацію, з якою буде йти робота. Показано принцип застосування атак на останній раунд шифру з внесенням збоїв у три різні позиції. Показано принцип застосування атак на проміжні раунди шифру з внесенням збоїв у три різні позиції. Пояснена різниця у застосуванні збоїв на діагональний і не діагональний елемент матриці байтів ключа. Наведено рівняння для пошуку кандидатів у байти ключа за збитим і звичайним шифротекстами і пояснено процес цього пошуку. Наведено оцінки складності пошуку кандидатів у байти ключа.

Ключові слова: АТАКА, ЗБІЙ, ДИФЕРЕНЦІАЛЬНИЙ АНАЛІЗ, QALQAN

ABSTRACT

The qualification work consists of: 60 pages, 5 figures, 5 tables, and 14 sources.

This work investigates the application of Differential Fault Analysis (DFA) on the modified Qalqan cipher, which is a candidate for the national standard of the Republic of Kazakhstan. The cipher and its modification, which will be the subject of this work, are described. The principle of applying attacks on the last round of the cipher with the introduction of faults in three different positions is demonstrated. The principle of applying attacks on intermediate rounds of the cipher with the introduction of faults in three different positions is shown. The difference in the application of faults on the diagonal and non-diagonal elements of the key byte matrix is explained. Equations for finding key byte candidates from faulty and regular ciphertexts are provided, and the process of this search is explained. Complexity estimates for finding key byte candidates are also provided.

Keywords: ATTACK, FAULT, DIFFERENTIAL ANALYSIS, QALQAN.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

DFA — диференційні атаки збоїв (англ. *differential fault attacks*)

SDFA — статистично-диференційні атаки збоїв (англ. *statistical-differential fault attack*)

$\circ$ — операція конкатенації (різних операцій або частин шифру)

$\oplus$ — операція побітового додавання (додавання за модулем 2)

$+$ — операція побайтового додавання без виходу за межі байту (додавання за модулем 2^8)

$-$ — операція побайтового віднімання без виходу за межі байту (віднімання за модулем 2^8).

$\boxplus$ — операція додавання за модулем 2^{128}.

$K^{(r)}$ — ключ, застосований у раундовій функції на раунді r.

$K_i^{(r)}$ — байт, який стоїть на i-й позиції ключа, що застосовано у раундовій функції на раунді r.

K_i — байт ключа, якщо з контексту зрозуміло, про який раунд йде мова.

$V_n = \{0,1\}^n$ — простір усіх двійкових векторів довижини n.

$V_n^8 = \{0,1,\ldots,9,A,B,C,D,E,F\}^n$ — простір усіх байтових векторів довжини n.

$a_{i,j}^K$ — елемент матриці з індексом i, j, на який було накладено ключ K.

$a_{i,j}^L$ — елемент матриці з індексом i, j, на який було застосовано лінійне перетворення L.

$a_{i,j}^S$ — елемент матриці з індексом i, j, на який було застосовано нелінійне перетворення S.

ВСТУП

Актуальність дослідження. На сьогодні активно розвивається розділ криптографії під назвою легка криптографія. Суть цього розділу полягає у розробці криптосистем, які працюють якомога швидше і використовують якомога менше ресурсів при цьому. Таким криптосистемам дозволяється понижена стійкість до класичних атак, в порівнянні з традиційними криптосистемами, через меншу кількість ресурсів. Однак такі криптосистеми мають бути стійкими до певної низки інших атак, які можна реалізувати використовуючи мінімальні обчислювальні ресурси.

До таких атак відносять атаки за сторонніми каналами, зокрема дифереціальні атаки збоїв (DFA). Концепт атак збоїв з'явився у 2001 році і швидко став одним з важливих критеріїв при побудові симетричних криптосистем у сфері легкої криптографії через свою простоту у реалізації і ефективність. На певну кількість сучасних шифрів було побудовано практичні атаки використовуючи цей підхід [9, 14, 5, 10].

Симетричний блоковий шифр Qalqan було запропоновано у 2021 році як кандидат на національний стандарт шифрування Республіки Казахстан [4]. Даний шифр є орієнтованим на байтову архітектуру. Qalqan позиціюється, як шифр легкої криптографії на основі архітектури SP-мережі. Для сучасних шифрів рівню національного стандарту вимагається стікість до атак на реалізацію, яка не була доведена у оригінальній роботі.

Метою дослідження є узагальнення диференціальних атак збоїв на блокові шифри спеціального виду. Для досягнення мети необхідно виконати такі **завдання дослідження**:

1) провести огляд опублікованих джерел за тематикою дослідження;
2) проаналізувати структуру та особливості шифру Qalqan;
3) побудувати атаку збоїв на ключ останнього раунду шифру Qalqan

та проаналізувати її ефективність;

4) побудувати атаку збоїв на ключі проміжних раундів шифру Qalqan та проаналізувати її ефективність.

Об'єктом дослідження є інформаційні процеси в системах захисту інформації.

Предметом дослідження є стійкість симетричного блокового шифру Qalqan до атак на реалізацію.

При розв'язанні поставлених завдань використовувались такі *методи дослідження*: методи лінійної алгебри, дискретної математики та математичної статистики, методи криптоаналізу.

Наукова новизна отриманих результатів полягає у тому, що вперше було запропоновано практичні атаки збоїв на шифр Qalqan.

Практичне значення результатів: одержані результати дозволяють сформулювати вимоги до захищених реалізацій шифру Qalqan.

Апробація результатів та публікації. Результати були представлені на II Міжнародній науково-практичній конференції «Кіберборотьба: розвідка, захист та протидія» (23-24 квітня 2024 р., м. Київ, Україна) [12], XXII Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» (13-17 травня 2024 р., м. Київ, Україна) [13].

1 ОСНОВНІ ПОНЯТТЯ І ПОПЕРЕДНІ РЕЗУЛЬТАТИ

У цьому розділі надається до розгляду аналіз сучасних трендів у захисті від DFA [7], розглядається поняття DFA, описується структура шифру DEFAULT, який був створений, як стійкий до DFA шифр [2], описані атаки Нагелера та ін. [8] і Джана та ін. [6] на шифр DEFAULT, сформульовані і розглянуті узагальнена стратегія атаки Нагелера та ін. на шифри з S-блоками з лінійними структурами та узагальнена стратегія атаки Джана та ін.. Розглянуто поняття SDFA та наведено основні принципи побудови атак цим методом на DEFAULT. Приводяться оцінки складності деяких з цих атак. Наведена структура шифру Qalqan, до якого буде застосована атака у подальших розділах. Пояснені модифікації, які будуть застосовані до шифру.

1.1 Опис ідеї побудови диференціальних атак збоїв

Ідея диференціальних атак збоїв з'явилось ще наприкінці 1990-х років, і сформувалась у 2001-му. Для розуміння того, як застосовуються диференціальні атаки збоїв на шифр Qalqan, необхідно розуміти, в чому полягає сутність диференціальних атак збоїв.

Означення 1.1 (DFA)**.** Диференціальні атаки збоїв (англ. *differential fault attacks*) — атака на імплементацію криптосистеми. За мету стоїть внесення збоїв у виконання криптографічних операцій для побудування залежностей і отримання інформації про ключ. Для цього вводять збої у біти і роблять спостереження про зміни результату, роблять висновки про можливі значення і повторюють експеримент для уточнення результатів.

– Внесення збою зазвичай означає зміну біта на протилежний, хоча також можна застосовувати занулення біту, що є іншою варіацією атаки.

– Для застосування потрібно чітко розуміти структуру раундових функцій, без цього аналіз неможливий.

– Структура раундових ключів також має вплив на атаку. Атакувати за допомогою DFA можна не лише раундові функції, а і генератори ключів.

– Важлива характеристика для аналізу: кількість бітів, які відрізняються у звичайному і збитому обчисленнях. Ця характеристика називається *диференціалом*, або *різницею входу*.

– Виділяють три основні характеристики складності диференціальних атак збоїв: кількість збоїв, які потрібно внести у виконання програми, результуюча кількість ключового простору і складність обчислення отриманих співвідношень.

Згідно дослідження Баксі та ін. [2], сучасні методи захисту від DFA включають наступні категорії (але не обмежуються ними):

1) Використання пристроїв, які помічають і намагаються уникнути виникненню збою, або захисні пристрої, які блокують можливість внесення збоїв.

2) Окремі протоколи безпеки на основі комунікації між Алісою та Бобом, які дозволяють з досить високою ймовірністю пересвідчитись у відсутності збоїв. Наприклад, для такого захисту можуть ввести захист у певну частину обчислювального пристрою.

3) Подвоєння шифрування з подальшою прихованою або явною перевіркою на однаковість виходів, яке називають *подвійним обчисленням*. До цієї категорії також відносять усі джерела надлишковості на рівні виконуючих пристроїв.

4) Використання математичних рішень для зниження ефективності DFA.

Баксі та ін. стверджують, що лише третя і четверта категорії даної класифікації підвладні розробнику шифра. Було зазначено, що для третьої категорії вже було винайдено атаки, які дозволяють це ускладнення подолати.

Шифр DEFAULT робить ставку на *стійкість до збоїв*, на відміну

від *виявлення збоїв*, які застосовуються у інших сучасних шифрах легкої криптографії. Суть стійкості полягає у тому, що зловмисник не дізнається достатньо змістовної інформації про ключовий простір при внесенні збоїв, і, як наслідок, не зможе отримати змістовного набору інформації для подальшого аналізу.

Введено поняття *лінійної структури*, яку Баксі та ін. використовують як підсилювач стійкості до DFA.

Лема 1.1. *Для входу* α *у S-блок* S*, якщо* $\alpha \oplus a$ *– розв'язок* $S(x) \oplus S(x \oplus \gamma) = \Delta(\alpha, \delta)$ *для усіх можливих різниць у входах* δ*, то* a $(\neq 0)$ *називаємо* лінійною структурою S.

Розглянемо шифр DEFAULT, який запропонували Баксі та ін. [2] для забезпечення стійкості від DFA у системах з обмеженими ресурсами.

1.2 Структура шифру DEFAULT

Шифр DEFAULT є SP-мережею і складається з двох основних частин: DEFAULT-LAYER та DEFAULT-CORE. У роботі Баксі та ін. було зазначено структуру для 128-бітної версію шифру.

Ці частини утворюють шифр за рахунок складання їх у наступну структуру: DEFAULT-LAYER ∘ DEFAULT-CORE ∘ DEFAULT-LAYER.

Спочатку розглянемо DEFAULT-LAYER. Його мета – захист шифру від DFA та усіх інший видів аналізу, які визначають відмінності між запусками. DEFAULT-LAYER складається з 28 раундів перестановки, яка приймає повідомлення довжини 128 бітів (або 32 4-бітові слова-тетради) і ключ довжини 128 бітів. Раундові функції R складаються з 4 послідовних операцій:

1) SubCells – застосування S-блоку до кожної тетради повідомлення. S-блок визначено як S = 037ED4A9CF18B265. Такий S-блок містить 4 лінійні структури.

2) PermBits – застосування перестановки P_{128}, яка напряму взята з

GIFT-128 [3].

3) AddRoundConstants – біт ’1’ і 6-бітну раундову константу $C = c_5c_4c_3c_2c_1c_0$ побітово додають на позиції 127, 23, 19, 15, 11, 7 і 3 відповідно.

4) AddRoundKey – до повідомлення після операції 3) побітово додається ключ.

У звичайній версії шифру, ключі на кожному раунді вважаються однаковими. У покращеній варіації DEFAULT було запропоновано використовувати 4 різні раундові ключі, що обертаються. Такі раундові ключі генеруються з наданого ключа K наступним чином:

$$K_0 = K, \quad K_{i+1} = R'(R'(R'(R'(K_i)))), i \in [0,1,2].$$

Тут R' – спрощена раундова функція з наступними змінами:

– Відсутній крок AddRoundKey;

– AddRoundConstants не робить побітового додавання раундової константи, а виключно додає біт ’1’ до найстаршого біту повідомлення.

Далі на раунді i використовується ключ $K_{i \bmod 4}$ в якості раундового.

Тепер опишемо структуру DEFAULT-CORE. Його мета – захист від більш традиційних атак, але зберігається структура дуже близька до DEFAULT-LAYER. Використовується 24 раундові функції з 4 послідовними операціями:

– SubCells – застосування S-блоку до кожної тетради повідомлення. S-блок визначено як S = 196F7C82AED043B5. Такий S-блок не містить лінійних структур.

– PermBits, AddRoundConstants, AddRoundKey – такі ж, як і у DEFAULT-LAYER.

DEFAULT-LAYER також пропонувався Баксі та ін., як окремий шар, який можна накладати на інші шифри для додавання захисту від DFA у них зі збереженням ефективності обчислень. Цей шар рекомендовано для розмірів блоку не менше 128 бітів, а також

запропоновано спосіб розширення його на довільну довжину кратну 16.

1.3 Атаки Нагелера та ін. на DEFAULT

Як було зазначено раніше, у DEFAULT існує як простий режим роботи, так і режим з ключами, що обертаються. Атаки Нагелера та ін. стосувались обох варіацій, але у цій роботі увага буде сфокусована саме атаках на режим з ключами, що обертаються, адже вони складніші і краще відповідають тематиці цієї роботи.

У своїй роботі Нагелер та ін. сфокусувались на пошуці додаткової інформації з S-блоків і раундових функцій шифру. Важливою частиною його аналізу є дослідження, які лінійні рівняння з використанням бітів певної тетради ключа можна дізнатись в залежності від кількості змін між звичайним і спотвореним входами у S-блок при шифруванні, або обернений S-блок у дешифруванні.

Було показано, що діставати додаткову інформацію з аналізу розповсюдження збоїв між раундами є можливим, що суперечить твердженню автора шифру. Таким чином, якщо під час аналізу вдалось обмежити кількість кандидатів значення тетради, то аналіз наступних раундів дозволяє зафіксувати якесь значення біта кандидата і розглядати решту, таким чином отримуючи зменшений ключовий простір.

Для аналізу ускладненого режиму роботи Нагелер та ін. розглянули еквівалентності, за якими можна розділити ключовий простір на класи і виділити нормалізовані ключі у кожному з класів для подальшого аналізу. Якщо використовувати однакові ключі на кожному раунді, або змінювати ключі лінійною функцією чи бітовою перестановкою, то у інформації про ключі будуть недостатньо сильні розсіювання. Іншою причиною виникнення такої можливості є введення лінійних структур у S-блоки. Це призведе до виникнення можливості об'єднати часткову інформацію про ключі у сусідніх раундах і, таким чином, визначити повний ключ.

Найбільш успішна атака використовує структуру атаки на декілька раундів, скомбіновану з узагальненою стратегію атаки Нагелера та ін. для шифрів з лінійними структурами, яку буде описано у наступному розділі і має наступну ідею:

Необхідно визначити по 64 біта з інформацією про кожен з 6 раундових ключів. Атака проводиться по 6 різним раундам дешифрування. Збої вставляються таким чином, щоб раундові функції обробили збій 4 рази перед додаванням з ключем, який прагнемо визначити. Збої вставляються у індекси від 1 до 29 включно з кроком 4. Таким чином, зміни відбудуться у 2 з 8 S-блоків у крайній правій позиції (з боку наймолодших бітів). Далі визначаються залежності і різниця між звичайним і збоїстим запуском S-блоку перед додаванням шуканого ключа. Далі фільтруємо тетради ключа, який прагнемо визначити по очікуваній різниці. Обмежуємо 6 ключів, які формують послідовність нормалізованих ключів таким чином, що тетради перших 5 ключів обмежені до {0,1,2,3}. Повторюємо збої у цих 8 S-блоках поки не отримаємо 64 біта з інформацією про ключ. На цьому кроці маємо 6 ключів, ключовий простір розміру 2^{64}. Коли зібрали достатньо інформації про 6 нормалізованих ключів $(\overline{K}_0, \overline{K}_1, \overline{K}_2, \overline{K}_3, \overline{K}_4, \overline{K}_5)$, переходимо від нормалізованих до не нормалізованих ключів. Ключовий простір розширюється до 2^{384} через відкидання нормалізації. Це дозволяє застосувати структуру ключа, що обертається і прирівняти $K_0 = K_4$, $K_1 = K_5$, що звузить ключовий простір до 2^{194}. Після цього знову додаємо умову, що перші 4 ключа мають утворювати нормалізовану послідовність раундових ключів і з системи лінійних рівнянь отримуємо рівно єдиний результат для ключа DEFAULT-LAYER, що обертається. Далі стандартними DFA отримуємо ключ DEFAULT-CORE. На цьому шифр можна вважати поламаним.

Важливою приміткою можна вважати те, що ця атака застосовна для довільної кількості незалежних ключів, якщо хоча б два ключі зустрічається більше ніж один раз. Конкретно для DEFAULT атака на x

ключів, що обертаються потребує 64 біта з інформацією про кожен з $x + 2$ ключів.

1.4 Узагальнена стратегія атаки Нагелера та ін. для шифрів з лінійними структурами

Спостереження про класи еквівалентності ключів дозволяють отримати $n - 1$ еквівалентний раундовий ключ за допомогою застосування DFA до $n - 1$ раунда з лінійними структурами і аналізу отриманого шифротексту. Розглянемо шифр з n незалежними раундовими ключами і $n - 1$ раундів цього шифру. При цьому n-ий ключ залишиться невідомим, якщо не вийде отримати його з інших частин шифру, бо після $n - 1$ раунду, в усі S-блоки буде внесено збій. Цю стратегію можна застосовувати як до процесу шифрування, так і розшифрування. У роботі була описана стратегія атаки для процесу розшифрування.

Нашою метою буде дістати $n - 1$ нормалізований раундовий ключ. Для цього вносимо один збій в кожен S-блок останнього раунда дешифрування і дізнаємось $\overline{K}_0$. Далі цей процес повторюється для $\overline{K}_1$, ..., $\overline{K}_{n-2}$. При цьому, коли отримується множина можливих значень нормалізованого ключа, можна обирати найменший з них в якості дійсного значення, оскільки інші можна легко виразити. Якщо різниця між справжнім ключем і ключем зі збоєм надто велика, її можна компенсувати у наступному раунді визначення ключа завдяки лінійним структурам.

Повертаючись до DEFAULT, виключно завдяки цій стратегії можна відновити 27 з 28 раундових ключів DEFAULT-LAYER, вносячи збої у сам DEFAULT-LAYER, і останній, внісши збої у DEFAULT-CORE. Але застосування тільки цієї стратегії не є настільки ефективною по кількості використаних збоїв, як атака, описана у попередньому розділі.

1.5 Атаки Джана та ін. на DEFAULT та її узагальнення

Атака Джана та ін. покладається на схожі концепції, як у атаці Нагелера та ін. Основною відмінністю є глибший аналіз залежностей, які можна дістати між раундами шифру.

Найбільш успішна атака використовує струтуру атаки на 5-й з кінця раунд з визначенням еквівалентних ключів. Спочатку вводимо у 5-й з кінця раунд 8 збоїв, завдяки цьому отримуємо принаймні 2 відмінності у 2-му з кінця і останньому раундах. Звідси отримуємо значення $\overline{K}_3$ та $\overline{K}_2$. Далі вводимо 16 збоїв у 7-й з кінця раунд і отримуємо значення $\overline{K}_1$. Завдяки отриманому при обчисленні $\overline{K}_3$, $\overline{K}_2$ п'ятираундному сліду можемо приблизно оцінити значення K_0. Ввівши 8 збоїв у 8-й з кінця раунд отримуємо повний еквівалентний ключ $(\overline{K}_0, \overline{K}_1, \overline{K}_2, \overline{K}_3)$. Можна помітити, що така стратегія не покладається на повторення ключів у раундових функціях, а отже, може бути застосована у ширшому спектрі випадків. Наприклад, Джана та ін. також навели узагальнення цієї атаки для довільної кількості раундів, але воно має дуже подібну структуру до узагальненої атаки Нагелера та ін. та описаної вище атаки на п'ятираундний слід з еквівалентними ключами.

Тепер опишемо узагальнену стратегію атаки Джана. Вставка двох збоїв у кожну тетраду в останній раунд шифрування зменшує ключовий простір тетрад з 2^4 до 2^2. Ітеративно можна обирати по одній тетраді ключа з кожної множини зменшених тетрад і отримувати $\overline{K}_3$ та $\overline{K}_2$. По цим даним можна дістати $\overline{K}_1$. При цьому, на 4-му з кінця раунді для тетради k_0 залишиться 2^2 варіантів. Для визначення нормалізованого ключа вводимо збої у попередні раунди і використовуємо інформацію про $\overline{K}_3$, $\overline{K}_2$ та $\overline{K}_1$ разом з детермінованим слідом обчислень до 4-х раундів з кінця. Тоді буде необхідно приблизно 256 збоїв для отримання єдиної трійки $\overline{K}_3$, $\overline{K}_2$ та $\overline{K}_1$ з 2^{64} варіантів. Найбільш ефективна атака у попередньому розділі застосовує цей підхід і націлена на зменшення

кількості необхідних збоїв.

1.6 SDFA на DEFAULT з ключем, що обертається та ідеальним ключем

У роботі Джана та ін. була запропонована атака, яка є новим потужним інструментом для атаки на шифри легкої криптографії.

Означення 1.2 (SDFA)**.** Статистично-диференціальна атака збоїв (англ. *Statistical-Differential Fault Attack*) - симбіоз двох існуючих видів атак: статистичних атак і диференціальних атак збоїв. Задачею є застосування ефективного статистичного аналізу збоїв (далі SFA), проти якого було винайдено контрміри, у комбінації з DFA для зменшення необхідної кількості збоїв.

– Додана можливість атакувати незалежний набір раундових ключів (буде описана у наступному розділі), чого не могли зробити, наприклад, атаки Нагелера та ін..

– Збої вносяться не у унікальні біти, а у спеціальні набори бітів (англ. *bit-set*). Це буде називатись «збій набору бітів».

– Атаки на простий, складний і «ідеальний» режими мають приблизно однакову складність у цій варіації атаки. Для складного і «ідеального» режимів це покращення над DFA в усіх варіаціях.

Джана та ін. запропонували вводити 4 збої набору бітів для отримання унікальної тетради ключа за допомогою SFA. Також було зазначено, що введення α збоїв набору бітів у тетраду зменшить ключовий просторі з 2^4 до $2^{4-\alpha}$.

Тепер розглянемо атаку на DEFAULT. Ключі, що обертаються приймають значення K_0, K_1, K_2, K_3. В останньому раунді DEFAULT-LAYER використовується раундовий ключ K_3. Вставивши три збої наборів бітів, в кожну тетраду останнього раунда, отримуємо можливість ефективно відновити значення k_3. Решту ключів відновлюємо

напряму за допомогою R'^{-1}.

Також була запропонована атака, яка використовує SDFA і при цьому працює на DEFAULT-LAYER у «ідеальному» варіанті: кожен раундовий ключ генерується незалежно, тобто знаючи один, нічого не можна сказати про інші. Атаки Нагелера та ін. і DFA вважаються можливими, але досить складними по кількості збоїв і часу. Для атаки Джана та ін. використовують 3 збої наборів бітів в кожен раунд DEFAULT-LAYER і, проаналізувавши отримані збої, отримати значення кожного раундового ключа. Ця атака є в x разів складнішою за описану у попередньому розділі, де x – кількість незалежних раундових ключів, але навіть з такою складністю вона, згідно оцінки Джана та ін., працює швидше за усі раніше запропоновані DFA.

1.7 Таблиця Джана та ін. з результатами атак

Таблиця 1.1 містить результати дослідження Джана та ін. [6]. На цій таблиці приведені складності у кількості збоїв і отримані ключові простори в результаті застосування різних стратегій атак Нагелера та ін. і Джана та ін. x позначає складність у збоях застосування стандартного DFA до не стійкого до DFA DEFAULT-CORE.

З цієї таблиці видно, що більшість атак на простий режим ключа працюють краще за режим з ключами, що обертаються. При цьому лише деякі концепції застосовні до «ідеального режиму», про що було зазначено у попередніх розділах.

1.8 Структура шифру Qalqan

Шифр Qalqan було запропоновано у 2021 році як кандидат на національний стандарт шифрування Республіки Казахстан [4]. Даний шифр є орієнтованим на байтову архітектуру, симетричним і блоковим у

Таблиця 1.1 – Результати дослідження Джана та ін. щодо стійкості DEFAULT до DFA

Режим ключа	Робота	Стратегія атаки	Результати		Посилання
			# збоїв	ключ. простір	
Простий	Нагелер та ін.	Enc-DEC IC-DFA	16	2^{39}	[8]
		Multi-round IC-DFA	16	2^{20}	
	Джана та ін.	Second-to-Last Round Attack	32	2^{32}	[6]
		Third-to-Last Round Attack	32	2^{0}	
		Fourth-to-Last Round Attack	16	2^{0}	
		Fifth-to-Last Round Attack	8	2^{0}	
		SDFA	$[64, 128]$	2^{0}	
Обертаючийся	Нагелер та ін.	Generic NK-DFA	$1728 + x$	2^{0}	[8]
		Enc-Dec IC-NK-DFA	$288 + x$	2^{32}	
		Multi-round IC-NK-DFA	$(84 \pm 15) + x$	2^{0}	
	Джана та ін.	Third-to-Last Round Attack	$96 + x$	2^{0}	[6]
		Fourth-to-Last Round Attack	$48 + x$	2^{0}	
		Fifth-to-Last Round Attack	$24 + x$	2^{0}	
		SDFA	[64,128]	2^{0}	

виконанні операцій.

– Розмір блоку складає 128 бітів;

– Довжина ключа може варіюватись від 256 до 1024 бітів з кроком у 128 бітів;

– В залежності від довжини ключа, схема шифрування містить від 17 до 23 раундів.

Шифр Qalqan побудовано на основі архітектури SP-мережі. У кожному раунді, окрім останнього, застосовується 3 основні операції: накладання раундового ключа, застосування нелінійного перетворення (S-блоку) та лінійного перетворення L, яке має унікальну структуру під назвою «Квадрат»; останній раунд містить тільки операцію накладання ключа (яке названо *маскуванням*). У лінійному перетворенні шифру використовується побайтове додавання + (тобто додавання за модулем 256 з переносом без виходу за межі байту). Ключі на усіх раундах, окрім першого і останнього, накладаються додаванням за модулем 2^{128} ($\boxplus$); у

першому та останньому раундах накладання ключа відбувається побітовим додаванням $\oplus$.

Існує декілька видів представлень блоків:

– вектором бітів $x \in V_{128}$ $(x_1,x_2,\ldots,x_{128})$;

– вектором байтів $x \in V_{16}^8$ $(x_1,x_2,\ldots,x_{16})$;

– матриця байтів розміру 4×4.

У даній роботі, усі блоки відкритого тексту, проміжні шифротексти та раундові ключі представляються матрицями розміру 4×4.

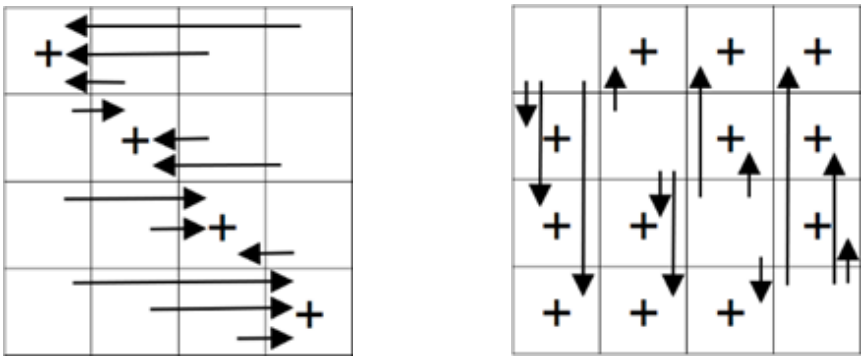
$$\begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{pmatrix} = \begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

Використовуватиметься довільний з двох варіантів позначення матриці, в залежності від зручності.

Лінійне перетворення $L : V_{128} \to V_{128}$ складається з фаз «всмоктування» і «розповсюдження». У формульному вигляді, у байтовому матричному представленні маємо:

$$\begin{aligned} a_{0,0}^L &= a_{0,0} + a_{0,1} + a_{0,2} + a_{0,3} \\ a_{1,0}^L &= a_{1,0} + a_{0,0}^L \\ a_{2,0}^L &= a_{2,0} + a_{0,0}^L \\ a_{3,0}^L &= a_{3,0} + a_{0,0}^L \\ a_{1,1}^L &= a_{1,0} + a_{1,1} + a_{1,2} + a_{1,3} \\ a_{0,1}^L &= a_{0,1} + a_{1,1}^L \\ a_{2,1}^L &= a_{2,1} + a_{1,1}^L \\ a_{3,1}^L &= a_{3,1} + a_{1,1}^L \\ a_{2,2}^L &= a_{2,0} + a_{2,1} + a_{2,2} + a_{2,3} \\ a_{0,2}^L &= a_{0,2} + a_{2,2}^L \\ a_{1,2}^L &= a_{1,2} + a_{2,2}^L \end{aligned}$$

$$a^L_{3,2} = a_{3,2} + a^L_{2,2}$$
$$a^L_{3,3} = a_{3,0} + a_{3,1} + a_{3,2} + a_{3,3}$$
$$a^L_{0,3} = a_{0,3} + a^L_{3,3}$$
$$a^L_{1,3} = a_{1,3} + a^L_{3,3}$$
$$a^L_{2,3} = a_{2,3} + a^L_{3,3}$$



(а) Фаза всмоктування **(б)** Фаза розповсюджування

Рисунок 1.1 – Схема роботи лінійного перетворення L

У роботі Яковлєва С. та Столовича М. [11] було виявлено, що запропонована структура шифру не є стійкою до диференціальних атак. Потенційно вразливими місцями є недостатньо якісно підібраний S-блок, слабке лінійне перетворення L з точки зору лавинних ефектів, додавання ключів за модулем 2^{128}, яке потенційно розповсюджує збої між бітами, і неточності у очікуваному авторами шифру розповсюдженні лавинних ефектів. З урахуванням зазначених слабкостей, для аналізу будуть використані модифікації шифру, які, згідно попереднім дослідженням, матимуть дещо кращу стійкість, у порівнянні з оригінальною версією.

У другому розділі цієї роботи використовується узагальнена версія шифру, яка містить r раундів з довжиною ключа 256 бітів. Ключі будуть накладатись, як у модифікації запропонованій [11], побайтовим додаванням без переносу за межі байту (тобто так, як додаються у перетворенні L) на усіх раундах, окрім першого і останнього. Збої будуть вноситись на рівні байтів з огляду на архітектуру модифікації.

У третьому розділі цієї роботи розглянуто узагальнену версію шифру, яка містить r раундів з ключем довжиною 256 бітів. Ключі на усіх раундах будуть накладатись побайтовим додаванням без переносу за межі байту. Збої будуть вносити на рівні байтів. Останній раунд r буде містити накладання ключа K_r, застосування нелінійного перетворення S-блоку, лінійного перетворення L, і заключне накладання ключа K_{r+1}.

Висновки до розділу 1

У цьому розділі було розглянуто шифр DEFAULT [2] та описано механізми, які закладались авторами на рівні шифру для захисту від DFA. Було описано сучасні методи захисту від DFA [7]. Було розглянуто атаки Нагелера та ін. [8] і Джана та ін. [6] з їх узагальненнями, які показали, що механізми захисту, запропоновані авторами шифру DEFAULT не працюють. Джана та ін. у своїй роботі стверджують, що жоден з запропонованих на сьогоднішній день варіант захисту від DFA не буде дієвим.

Автори припускали, що додавання у S-блоки лінійних структур дозволить достатньо ускладнити пошук залежностей у шифрі, щоб DFA стало неефективним, зменшуючи ключовий простір з 2^n лише до $2^{\frac{n}{2}}$ варіантів для n-бітного блокового шифру. Як Нагелер та ін., так і Джана та ін. змогли добути єдиний ключ, що повністю спростовує твердження авторів DEFAULT. При цьому, обраний підхід до атаки сильно впливав на складність як по кількості збоїв, так і по отриманому ключовому простору. Найкращі атаки, які були розглянуті у розділі 1, потребують менше 100 збоїв і дають єдиний ключ. Застосування SDFA до набору незалежних ключів не продемонстровано у таблиці, але його результат можна вважати найкращим на сьогоднішній день для задач DFA такої складності. Описані Нагелером та ін. атаки в найбільш ефективному їх варіанти мають ймовірність успіху 95% і вище зі складністю, близькою до описаної. Джана та ін. стверджують, що його методи атаки можна

вважати детерміновані, з єдиною змінною - кількістю збоїв, які треба буде ввести.

Джана та ін. також показали, що введення інших шифрів замість DEFAULT-CORE, як, наприклад, BAKSHEESH [1], все одно не дасть можливості захистись від DFA.

Було розглянуто концепцію диференціальних атак збоїв, які вже використовувались для побудови ефективних атак на сучасні шифри легкої криптографії. Також показано структуру шифра Qalqan, з яким буде йти робота, і описано його модифікації. На шифр Qalqan ще не було побудовано жодної ефективної атаки, використовуючи метод диференціальних атак збоїв, тому буде запропоновано деякі варіанти у подальших розділах.

2 АТАКИ НА КЛЮЧ ОСТАННЬОГО РАУНДУ ШИФРУ QALQAN

У цьому розділі показано принцип застосування атак на останній раунд модифікованого шифру Qalqan. Метою аналізу є отримання ключа $K^{(r)}$. Для цього розглядається внесення збоїв у раунди $r-1$, $r-2$ та $r-3$ і аналіз отриманих диференціалів.

2.1 Атака з внесенням збоїв у раунд $r-1$

Для проведення атаки важливо розуміти якість розповсюдження збоїв у функції. S-блок шифру не впливає на кількість збитих байтів, як і накладання ключа додаванням за модулем $2^8(+)$, згідно оголошених модифікацій. Для лінійного перетворення L позиція збитого байту є важливою, адже байт, який у матриці не стоїть на діагоналі, буде розповсюджено на п'ять байтів замість чотирьох. Розглянемо створені концепти атак.

Атакою на раунд $r-1$ називаємо послідовність дій:

– Внесення збою у певний байт матриці проміжного тексту перед раундом $r-1$;

– Накладання ключа $K^{(r-1)}$ побайтовим додаванням $+$;

– Застосування нелінійного перетворення S-блоку;

– Застосування лінійного перетворення L;

– Накладання ключа $K^{(r)}$ побітовим додаванням $\oplus$ (маскування);

– Порівняння отриманих збитого і звичайного шифротекстів;

– Отримання необхідної інформації про залежності і виділення кандидатів у байти ключа.

Атака на останній раунд дає можливість дізнатись початкові залежності у диференціалах між збитими і звичайними шифротекстами

при застосуванні раундової функції і побачити певні тенденції у розповсюдженні збоїв. Збої у байти вносимо додаванням по модулю 2^8 до обраного байту матриці випадкового значення збою. Інформацію для відновлення байтів ключа отримуємо з порівняння збитих шифротекстів і звичайних і розв'язку утворених співвідношень.

2.1.1 Внесення збою у діагональний елемент

На рисунку 2.1 можна побачити усі можливі варіанти лавинних ефектів при одиничному застосуванні лінійного перетворення L до збитого байту.

Позначимо збій, який ми внесли у довільний діагональний елемент, як α. Покажемо, як змінюється матриця байтів:

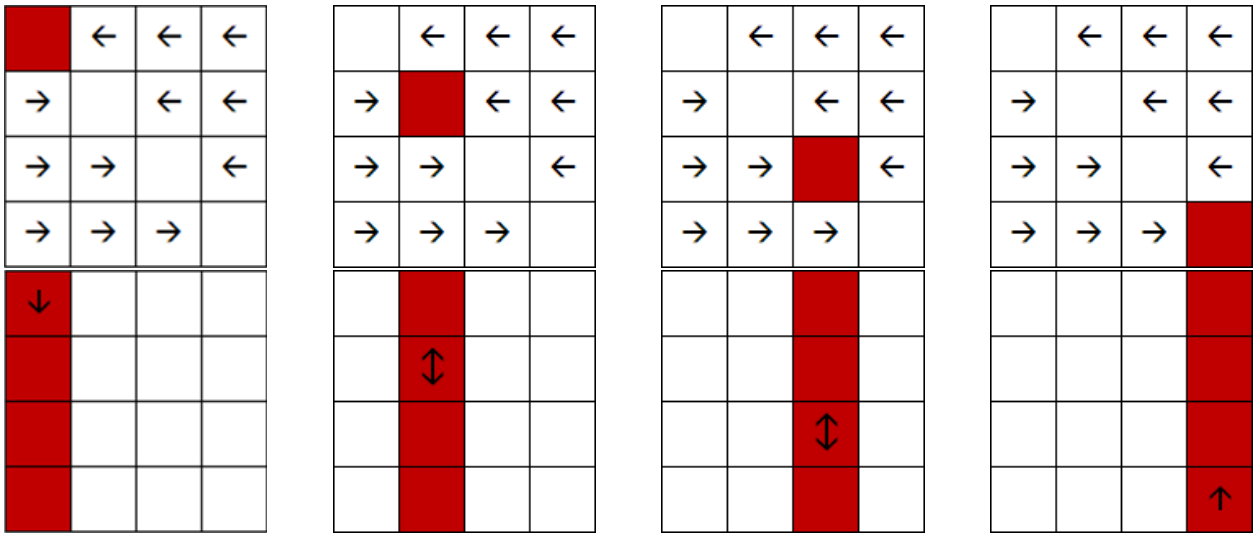
Нехай задано початкову матрицю на даному етапі шифру

$$\begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

Після внесення збою отримуємо

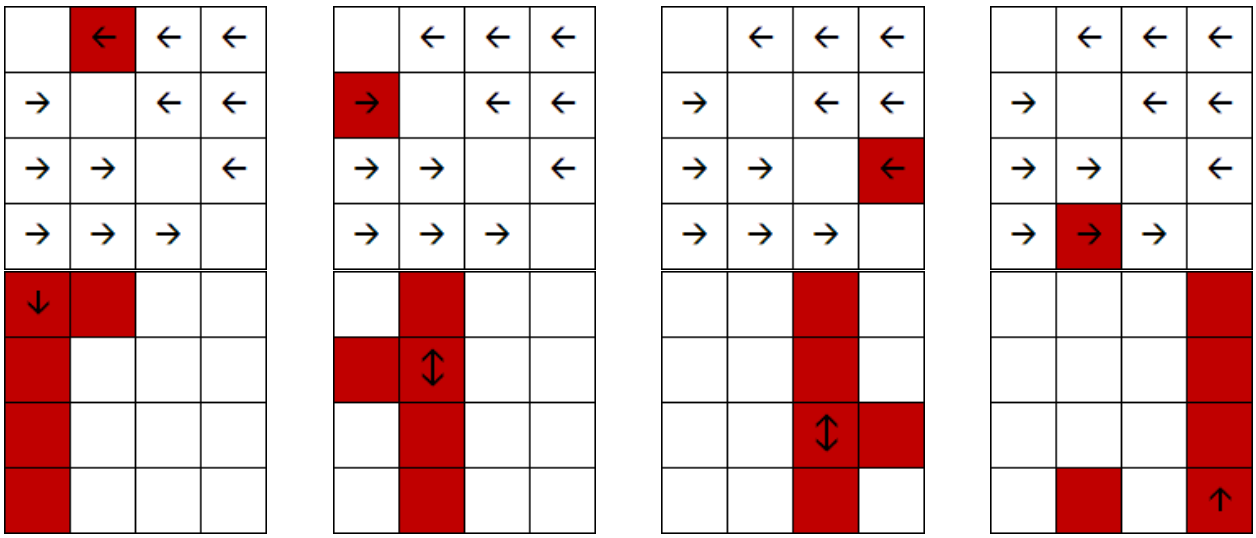
$$\begin{pmatrix} a_{0,0} + \alpha & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

Накладання ключа можна предствити у вигляді додавання елементів



(а) Збито байт $a_{0,0}$ **(б)** Збито байт $a_{1,1}$ **(в)** Збито байт $a_{2,2}$ **(г)** Збито байт $a_{3,3}$

Рисунок 2.1 – Мапи лавинних ефектів при збитті байту на діагоналі матриці при застосуванні лінійного перетворення L



(а) Збито байт $a_{0,1}$ **(б)** Збито байт $a_{1,0}$ **(в)** Збито байт $a_{2,3}$ **(г)** Збито байт $a_{3,1}$

Рисунок 2.2 – Мапи лавинних ефектів при збитті деяких байтів не на діагоналі матриці при застосуванні лінійного перетворення L

$K_{i,j}^{(r-1)}$ до відповідних байтів. Позначимо $a_{i,j}^K := a_{i,j} + K_{i,j}^{(r-1)}$

$$\begin{pmatrix} a_{0,0}^K + \alpha & a_{0,1}^K & a_{0,2}^K & a_{0,3}^K \\ a_{1,0}^K & a_{1,1}^K & a_{1,2}^K & a_{1,3}^K \\ a_{2,0}^K & a_{2,1}^K & a_{2,2}^K & a_{2,3}^K \\ a_{3,0}^K & a_{3,1}^K & a_{3,2}^K & a_{3,3}^K \end{pmatrix}$$

Бачимо, що збій зберігся без змін. Застосуємо S-блок до кожного байту. Позначимо $a_{i,j}^{KS} := S(a_{i,j}^K)$. Оскільки застосування S-блоку нелінійне, то позначимо $a_{0,0}^{KS} + \beta := S(a_{i,j}^K + \alpha)$, оскільки значення збитого результату застосування S-блоку і звичайного буде також відрізнятись на деяке число β за модулем 2^8.

$$\begin{pmatrix} a_{0,0}^{KS} + \beta & a_{0,1}^{KS} & a_{0,2}^{KS} & a_{0,3}^{KS} \\ a_{1,0}^{KS} & a_{1,1}^{KS} & a_{1,2}^{KS} & a_{1,3}^{KS} \\ a_{2,0}^{KS} & a_{2,1}^{KS} & a_{2,2}^{KS} & a_{2,3}^{KS} \\ a_{3,0}^{KS} & a_{3,1}^{KS} & a_{3,2}^{KS} & a_{3,3}^{KS} \end{pmatrix}$$

Застосуємо перетворення L. Як було описано раніше, $a_{i,i}^{KSL} = +_j \ a_{i,j}^{KS}$, $a_{i,j}^{KSL} = a_{i,j}^{KS} + a_{j.j}^{KSL}$, $i \neq j$

$$\begin{pmatrix} a_{0,0}^{KSL} + \beta & a_{0,1}^{KSL} & a_{0,2}^{KSL} & a_{0,3}^{KSL} \\ a_{1,0}^{KSL} + \beta & a_{1,1}^{KSL} & a_{1,2}^{KSL} & a_{1,3}^{KSL} \\ a_{2,0}^{KSL} + \beta & a_{2,1}^{KSL} & a_{2,2}^{KSL} & a_{2,3}^{KSL} \\ a_{3,0}^{KSL} + \beta & a_{3,1}^{KSL} & a_{3,2}^{KSL} & a_{3,3}^{KSL} \end{pmatrix}$$

Накладання ключа через $\oplus$ ніяк не вплине на подальше

розповсюдження чи значення збою. Відповідно диференціал між нормальним і збитим шифротекстом буде

$$\begin{pmatrix} \beta & 0 & 0 & 0 \\ \beta & 0 & 0 & 0 \\ \beta & 0 & 0 & 0 \\ \beta & 0 & 0 & 0 \end{pmatrix}$$

Позначимо лівий стовпчик нормального ШТ за y_1, y_2, y_3, y_4, тоді лівий стовпчик збитого ШТ буде $y'_i = y_i + \beta$.

Знаючи це, можемо утворити наступні рівняння:

$$(y_1 \oplus K_1^{(r)}) - (y'_1 \oplus K_1^{(r)}) = (y_2 \oplus K_2^{(r)}) - (y'_2 \oplus K_2^{(r)}) \tag{2.1}$$

$$(y_3 \oplus K_3^{(r)}) - (y'_3 \oplus K_3^{(r)}) = (y_4 \oplus K_4^{(r)}) - (y'_4 \oplus K_4^{(r)}) \tag{2.2}$$

$$(y_1 \oplus K_1^{(r)}) - (y'_1 \oplus K_1^{(r)}) = (y_3 \oplus K_3^{(r)}) - (y'_3 \oplus K_3^{(r)}) \tag{2.3}$$

$$(y_1 \oplus K_1^{(r)}) - (y'_1 \oplus K_1^{(r)}) = (y_4 \oplus K_4^{(r)}) - (y'_4 \oplus K_4^{(r)}) \tag{2.4}$$

$$(y_2 \oplus K_2^{(r)}) - (y'_2 \oplus K_2^{(r)}) = (y_3 \oplus K_3^{(r)}) - (y'_3 \oplus K_3^{(r)}) \tag{2.5}$$

$$(y_2 \oplus K_2^{(r)}) - (y'_2 \oplus K_2^{(r)}) = (y_4 \oplus K_4^{(r)}) - (y'_4 \oplus K_4^{(r)}) \tag{2.6}$$

За допомогою брутфорсу, можна перебрати значення $K_1^{(r)}$ та $K_2^{(r)}$ (2^{16} варіантів пар ключів) знаходимо усі значення, для яких виконується рівняння (2.1). Фіксуємо ці значення, повторюємо для y_3, y'_3 та y_4, y'_4 і отримуємо кандидати у пари ключів $K_3^{(r)}$, $K_4^{(r)}$, які задовольняють рівнянню (2.2).

Отримані значення називаємо *кандидатами у байти ключа*. Задачею є зменшення кількості кандидатів до мінімально можливої. Для зменшення застосовуємо *крос-валідацію*, суть якої полягає у повторі пошуку розв'язків для рівнянь (2.3) - (2.6) і перетині множин отриманих кандидатів.

2.1.2 Внесення збою у не діагональний елемент

На рисунку 2.2 можна побачити декілька можливих варіантів лавинних ефектів при одиничному застосуванні лінійного перетворення L до збитого байту, який стоїть не на діагоналі матриці. Усі інші випадки збитих елементів не на діагоналі будуть відрізнятись лише тим, який байт, на додачу до стовпця, буде збито.

Збій у не діагональному елементі матриці байтів матиме дуже схожий вигляд на збій у діагональному елементі, тому пропустимо кроки аж до лінійного перетворення L і розглянемо як відбудеться розповсюдження. Аналогічно до попереднього випадку, на цьому кроці ми матимемо $a^{KSL}_{i,i} = +_j\ a^{KS}_{i,j}$, $a^{KSL}_{i,j} = a^{KS}_{i,j} + a^{KSL}_{j.j}$, $i \neq j$. Також позначимо збій (змінений S-блоком, тобто різницю між застосуванням S-блоку до звичайного байту і збитого), як β. Нехай збитим байтом був елемент $a_{2,3}$. Його буде всмоктано у $a_{2,2}$ і розповсюджено по 3-му стовпцю.

$$\begin{pmatrix} a^{KSL}_{0,0} & a^{KSL}_{0,1} & a^{KSL}_{0,2} + \beta & a^{KSL}_{0,3} \\ a^{KSL}_{1,0} & a^{KSL}_{1,1} & a^{KSL}_{1,2} + \beta & a^{KSL}_{1,3} \\ a^{KSL}_{2,0} & a^{KSL}_{2,1} & a^{KSL}_{2,2} + \beta & a^{KSL}_{2,3} + \beta \\ a^{KSL}_{3,0} & a^{KSL}_{3,1} & a^{KSL}_{3,2} + \beta & a^{KSL}_{3,3} \end{pmatrix}$$

Після накладання ключа, отриманий диференціал матиме вигляд:

$$\begin{pmatrix} 0 & 0 & \beta & 0 \\ 0 & 0 & \beta & 0 \\ 0 & 0 & \beta & \beta \\ 0 & 0 & \beta & 0 \end{pmatrix}$$

Бачимо, що ми отримали на один диференціал більше. Цей додатковий збитий елемент можна позначити, як y_5, а відповідний йому ключ, як $K_5^{(r)}$. Далі ці нові елементи можна буде використати для утворення додаткових рівнянь. Наведемо приклад рівняння, яке можна утворити:

$$(y_3 \oplus K_3^{(r)}) - (y_3' \oplus K_3^{(r)}) = (y_5 \oplus K_5^{(r)}) - (y_5' \oplus K_5^{(r)})$$

Для пошуку інших кандидатів у випадку збою не діагонального елементу можна застувати рівняння, описані у розділі 2.1.1. Також, наявність п'ятого елементу дозволяє утворити більшу кількість уточнюючих рівнянь для крос-валідації. Таким чином, можна зробити висновок, що введення збою у не діагональний елемент буде ефективніше з точки зору зменшення ключового простору за один внесений збій. Додавання нового рівняння дозволить за рахунок витрати більшого часу на обчислення отримати кращу точність завдяки додатковій крос-валідації з ще одним елементом.

Експериментально перевірено, що повторне застосування випадкового збою у однаковому місці дозволяють майже напевне отримати єдиного кандидата у байти ключа, ймовірність не отримати однозначне значення байту ключа знехтовно мала при третьому застосуванні збою.

Опишемо узагальнений алгоритм атаки з внесенням збоїв у раунд $r-1$:

1) Обираємо довільний відкритий текст X.

2) Зашифровуємо X за допомогою шифру Qalqan, отримуємо звичайний шифротекст Y.

3) Зашифровуємо X за допомогою шифру Qalqan, але у раунді $r-1$ вносимо збій у байт x_1, отримуємо збитий шифротекст Y'.

4) Розв'язуємо рівняння (2.1) для отримання кандидатів у значення байтів ключа $K_1^{(r)}$ та $K_5^{(r)}$, та (2.2) для отримання кандидатів у значення

байтів ключа $K_9^{(r)}$ та $K_{13}^{(r)}$.

5) Повторюємо крок 3 з рівняннями (2.3) - (2.6) для уточення можливого значення кандидатів у ці байти ключа.

6) Повторюємо кроки 3-5 тричі для максимального уточнення значення байту ключа за допомогою крос-валідації.

7) Повторюємо кроки 3-6, змінивши місце, в яке вноситься збій на x_6, отримуємо байти ключа $K_2^{(r)}$, $K_6^{(r)}$, $K_{10}^{(r)}$, $K_{14}^{(r)}$.

8) Повторюємо кроки 3-6, змінивши місце, в яке вноситься збій на x_{11}, отримуємо байти ключа $K_3^{(r)}$, $K_7^{(r)}$, $K_{11}^{(r)}$, $K_{15}^{(r)}$.

9) Повторюємо кроки 3-6, змінивши місце, в яке вноситься збій на x_{16}, отримуємо байти ключа $K_4^{(r)}$, $K_8^{(r)}$, $K_{12}^{(r)}$, $K_{16}^{(r)}$.

10) Маючи усі значення байтів ключа, відновлюємо його.

У цьому алгоритмі можна пропускати кроки 5 і 6, але від цього обмеження простору ключів буде значно зменшено. Крок 6 може бути пропущено з огляду на можливість існування реалізації, яка не дозволяє внести збій більше, ніж 1 раз у одне місце.

Оціночні значення складності атаки зазначено на таблиці 2.1. Складність у диференційних атаках збоїв складається з кількості збоїв, які потрібно внести у виконання програми, результуючої кількості кандидатів у ключі і складність обчислення отриманих співвідношень (брутфорс рівнянь).

Таблиця 2.1 – Оціночні значення складності атаки з вносом збою у раунд $r-1$

# збоїв	Складність розв'язку рівнянь	# кандидатів у ключі (в середньому)
4	$24 \cdot 2^{16}$	2^{64}
8	$48 \cdot 2^{16}$	$\approx 2^0$
12	$72 \cdot 2^{16}$	2^0

2.2 Атака з внесенням збоїв у раунд $r-2$

У диференціальному аналізі збоїв розглядається внесення збоїв у різні раунди з метою отримання додаткової інформації про байти і нових залежностей. Збої будемо вносити побайтово, як і у попередній атаці. Метою є отримання іншої інформації про кандидати у байти ключа з порівняння збитого шифротексту і звичайного.

Атака на передостанній раунд складається з послідовності дій:

– Внесення збою у певний байт матриці

– Накладання ключа $K^{(r-2)}$ побайтовим додаванням +

– Застосування нелінійного перетворення S-блоку.

– Застосування лінійного перетворення L.

– Накладання ключа $K^{(r-1)}$ побайтовим додаванням +

– Застосування нелінійного перетворення S-блоку.

– Застосування лінійного перетворення L.

– Накладання ключа $K^{(r)}$ побітовим додаванням $\oplus$ (маскування).

– Порівняння отриманих збитого і звичайного шифротекстів;

– Отримання необхідної інформації про залежності і виділення кандидатів у байти ключа.

Суттєвою відмінністю між атакою на передостанній і останній раунди з точки зору збоїв є подвійне застосування лінійного перетворення L. Оскільки це єдина операція, яка розповсюджує збиті байти, необхідно звернути увагу на результуючі лавинні ефекти. Малюнки міститимуть лише один приклад лавинного ефекту для обох випадків, але у межах цих випадків інші лавинні ефекти матимуть дуже схожий вигляд з точністю до перестановок місцями стовпчиків. Також необхідно зазначити, що накладання ключа $K^{(r-1)}$ та застосування S-блоку призводить до зміни значення збоїв, отриманих після першого застосування лінійного перетворення L на інші, невідомі атакуючому значення. Це призводить до необхідності введення нових змінних для позначення диференціалів і

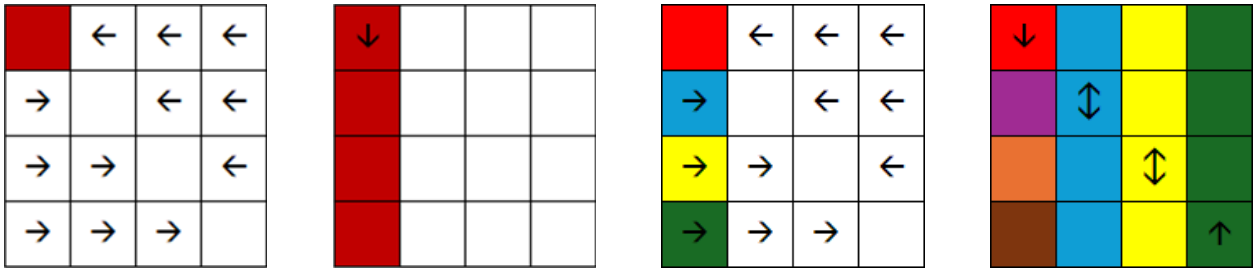
виникнення нових залежностей.

2.2.1 Внесення збою у діагональний елемент

Розглянемо лавинний ефект на прикладі рисунку 2.3. Позначимо збій, який розповсюджується завдяки дії першого перетворення L, як ω. Після накладання ключа $K^{(r-1)}$ та застосування S-блоку отримаємо чотири різні збої (як і раніше, з точки зору різниці між результатом застосування S-блоку до збитого і нормального шифротекстів). Якщо розглядати випадок, коли збитим бітом був $a_{0,0}$, то збої будуть на позиціях $a_{i,0}, i = \overline{0,3}$. Позначимо їх $\alpha, \beta, \gamma, \delta$. Початковий вигляд диференціалу і збитих тексів буде аналогічний до описаного у розділі 2.1.1. Застосуємо друге лінійного перетворення L і покажемо, як поширюються диференціали:

$$\begin{pmatrix} a_{0,0}+\alpha & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0}+\beta & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0}+\gamma & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0}+\delta & a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

$$\begin{pmatrix} a_{0,0}+\alpha & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0}+\beta & a_{1,1}+\beta & a_{1,2} & a_{1,3} \\ a_{2,0}+\gamma & a_{2,1} & a_{2,2}+\gamma & a_{2,3} \\ a_{3,0}+\delta & a_{3,1} & a_{3,2} & a_{3,3}+\delta \end{pmatrix}$$



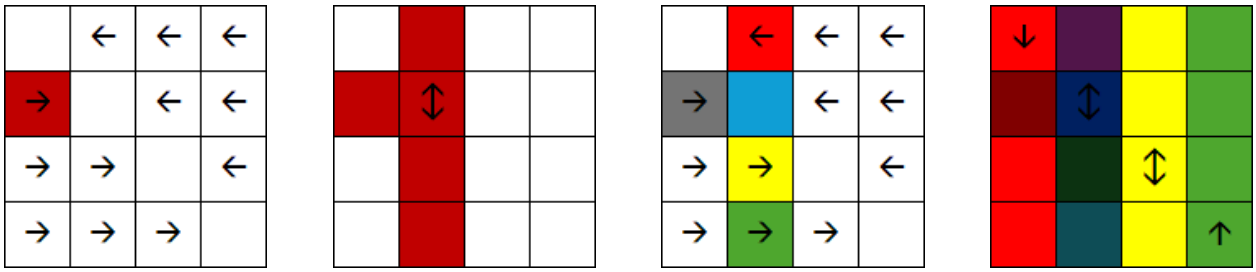
(а) Збито байт $a_{0,1}$ і всмоктано елементи

(б) Збій розповсюдився на стовпчик

(в) Значення збою змінилось і всмоктано байти

(г) Поширення збою на усі байти матриці

Рисунок 2.3 – Мапа лавинного ефекту при збитті байту на діагоналі матриці байтів.



(а) Збито байт $a_{1,0}$ і всмоктано байти

(б) Збій розповсюдився на стовпчик

(в) Значення збою змінилось і всмоктано байти

(г) Поширення збою на усі байти матриці

Рисунок 2.4 – Мапа лавинного ефекту при збитті байту не на діагоналі матриці байтів.

$$\begin{pmatrix} a_{0,0}+\alpha & a_{0,1}+\beta & a_{0,2}+\gamma & a_{0,3}+\delta \\ a_{1,0}+\alpha+\beta & a_{1,1}+\beta & a_{1,2}+\gamma & a_{1,3}+\delta \\ a_{2,0}+\alpha+\gamma & a_{2,1}+\beta & a_{2,2}+\gamma & a_{2,3}+\delta \\ a_{3,0}+\alpha+\delta & a_{3,1}+\beta & a_{3,2}+\gamma & a_{3,3}+\delta \end{pmatrix}$$

Після накладання ключа, отриманий диференціал матиме вигляд:

$$\begin{pmatrix} \alpha & \beta & \gamma & \delta \\ \alpha+\beta & \beta & \gamma & \delta \\ \alpha+\gamma & \beta & \gamma & \delta \\ \alpha+\delta & \beta & \gamma & \delta \end{pmatrix}$$

Спочатку розглянемо другий стовпчик, який містить тільки диференціал β. Утворимо рівнняння аналогічні до (2.1), і з цього дізнаємось $K_{i,1}^{(r)}, i = \overline{0,3}$. Відтворимо аналогічну процедуру для третього і четвертого стовпчиків, і отримаємо $K_{i,2}^{(r)}, i = \overline{0,3}$, $K_{i,3}^{(r)}, i = \overline{0,3}$. Віднявши збитий шифротекст від звичайного, можна визначити значення диференціалів β, γ, δ. Отримавши ці значення, можна підставити їх у перший стовпчик і звести значення диференіалу до α. Повторивши пошук кандидатів, можна визначити $K_{i,1}^{(r)}, i = \overline{0,3}$. Таким чином можна визначити кандидатів на усі байти ключа раунду r. Якщо кандидатів буде забагато, повторюємо збій у тому ж місці, що дозволить отримати нові значення $\alpha, \beta, \gamma, \delta$ і провести крос-валідацію.

2.2.2 Внесення збою у не діагональний елемент

Розглянемо лавинний ефект на прикладі рисунку 2.4. Позначимо збій, який розповсюджується завдяки дії першого перетворення L, як ω. Після накладання ключа $K^{(r-1)}$ та застосування S-блоку отримаємо п'ять різних збоїв (як і раніше, з точки зору різниці між результатом застосування S-блоку до збитого і нормального шифротекстів). Якщо розглядати випадок, коли збитим бітом був $a_{1,0}$, то збої будуть на позиціях $a_{1,0}, a_{i,1}, i = \overline{0,3}$. Позначимо їх $\alpha, \beta, \gamma, \delta, \epsilon$. Застосуємо друге лінійного перетворення L і покажемо, як поширюються диференціали:

$$\begin{pmatrix} a_{0,0} & a_{0,1}+\beta & a_{0,2} & a_{0,3} \\ a_{1,0}+\alpha & a_{1,1}+\gamma & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1}+\delta & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1}+\epsilon & a_{3,2} & a_{3,3} \end{pmatrix}$$

$$\begin{pmatrix} a_{0,0}+\beta & a_{0,1}+\beta & a_{0,2} & a_{0,3} \\ a_{1,0}+\alpha & a_{1,1}+\alpha+\gamma & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1}+\delta & a_{2,2}+\delta & a_{2,3} \\ a_{3,0} & a_{3,1}+\epsilon & a_{3,2} & a_{3,3}+\epsilon \end{pmatrix}$$

$$\begin{pmatrix} a_{0,0}+\beta & a_{0,1}+\alpha+\beta+\gamma & a_{0,2}+\delta & a_{0,3}+\epsilon \\ a_{1,0}+\alpha+\beta & a_{1,1}+\alpha+\gamma & a_{1,2}+\delta & a_{1,3}+\epsilon \\ a_{2,0}+\beta & a_{2,1}+\alpha+\gamma+\delta & a_{2,2}+\delta & a_{2,3}+\epsilon \\ a_{3,0}+\beta & a_{3,1}+\alpha+\gamma+\epsilon & a_{3,2}+\delta & a_{3,3}+\epsilon \end{pmatrix}$$

Після накладання ключа, отриманий диференціал матиме вигляд:

$$\begin{pmatrix} \beta & \alpha+\beta+\gamma & \delta & \epsilon \\ \alpha+\beta & \alpha+\gamma & \delta & \epsilon \\ \beta & \alpha+\gamma+\delta & \delta & \epsilon \\ \beta & \alpha+\gamma+\epsilon & \delta & \epsilon \end{pmatrix}$$

Спочатку розглянемо третій стовпчик, який містить тільки диференціал δ. Утворимо рівняння аналогічне до описаного у розділі (2.1), і з цього дізнаємось $K_{i,2}^{(r)}, i = \overline{0,3}$. Зробимо аналогічно з четвертим стовпчиком, отримаємо $K_{i,3}^{(r)}, i = \overline{0,3}$. Аналогічно з першого стовпчика можна отримати $K_{0,0}^{(r)}$, $K_{2,0}^{(r)}$, $K_{3,0}^{(r)}$. По різниці між звичайним і збитим шифротекстом визначаємо значення значення β, δ, ϵ. Підставляємо значення β у елемент $a_{1,0}$ і отримуємо елемент, який залежить лише від α. Далі значення диференціалів використовуємо для спрощення другого ствопчику. Виконуємо перебір значень збою α, дізнаємось значення $K_{1,0}^{(r)}$, $K_{i,1}^{(r)}, i = \overline{0,3}$. Як і у випадку 2.2.1, якщо кількість кандидатів надто велика, робимо уточнення крос-валідацією. Якщо крос-валідації не достатньо для уточнення результатів, робимо новий запуск з новим збоєм у тому ж місці і пошук кандидатів.

Опишемо узагальнений алгоритм атаки з внесенням збоїв у раунд $r-2$:

1) Обираємо довільний відкритий текст X.

2) Зашифровуємо X за допомогою шифру Qalqan, отримуємо звичайний шифротекст Y.

3) Зашифровуємо X за допомогою шифру Qalqan, але у раунді $r-2$ вносимо збій у байт x_1, отримуємо збитий шифротекст Y'.

4) Розв'язуємо рівняння (2.1), (2.2) на кожному з трьох стовпчиків, який не містить збою α, отримуємо кандидатів у значення байтів ключа r: $K_i^{(r)}$, де i це усі індекси матриці, окрім тих, які містять збій α.

5) Уточнюємо результати за допомогою розв'язку рівнянь (2.3) - (2.6).

6) Повторюємо кроки 3-5 тричі для максимального уточнення значення байту ключа за допомогою крос-валідації.

7) Повторюємо кроки 3-6, змінивши місце, в яке вноситься збій на довільний діагональний, який не знаходиться у першому рядку, отримуємо байти ключа, які до цього не розглядались, адже вони містились у стовпчику, який містив збій α.

8) Маючи усі значення байтів ключа, відновлюємо його.

У цьому алгоритмі можна пропускати кроки 5 і 6, але від цього обмеження простору ключів буде значно зменшено. Крок 6 може бути пропущено з огляду на можливість існування реалізації, яка не дозволяє внести збій більше, ніж 1 раз у одне місце.

Оціночні значення складності атаки зазначено на таблиці 2.2. Складність у диференційних атаках збоїв складається з кількості збоїв, які потрібно внести у виконання програми, результуючої кількості кандидатів у ключі і складність обчислення отриманих співвідношень (брутфорс рівнянь).

Таблиця 2.2 – Оціночні значення складності атаки з вносом збою у раунд $r - 2$

# збоїв	Складність розв'язку рівнянь	# кандидатів у ключі (в середньому)
2	$24 \cdot 2^{16}$	2^{64}
4	$48 \cdot 2^{16}$	$\approx 2^0$
6	$96 \cdot 2^{16}$	2^0

2.3 Аналіз диференціалів при внесенні збоїв у раунд $r - 3$

Аналіз структури раундових функцій показав, що внесення збоїв у раунд $r - 3$ в будь-який байт призведе до зміни усіх байтів перед

застосуванням останньої раундової функції. Після застосування двох раундових функцій кожен байт буде мати таку структуру, як описано у розділі з атакою на передостанній раунд. Після накладання ключа $K^{(r-1)}$ і застосування S-блоку кожен байт отримає унікальні значення диференціалів, які потім поширюються за допомогою лінійного перетворення L. Таким чином, немає необхідності розглядати внесення збою у діагональний чи недіагональний елементи окремо.

Диференціали для атаки на раунд $r-3$ отримаємо у результаті наступні:

$$\begin{pmatrix} \alpha & \beta & \gamma & \delta \\ \epsilon & \zeta & \eta & \theta \\ \iota & \kappa & \lambda & \mu \\ \nu & \xi & o & \pi \end{pmatrix} \rightarrow \begin{pmatrix} \alpha+\beta+\gamma+\delta & \beta & \gamma & \delta \\ \epsilon & \zeta+\epsilon+\eta+\theta & \eta & \theta \\ \iota & \kappa & \lambda+\iota+\kappa+\mu & \mu \\ \nu & \xi & o & \pi+\nu+\xi+o \end{pmatrix}$$

$$\begin{pmatrix} \alpha+\beta+\gamma+\delta & \beta+\zeta+\epsilon+\eta+\theta & \gamma+\lambda+\iota+\kappa+\mu & \delta+\pi+\nu+\xi+o \\ \epsilon+\alpha+\beta+\gamma+\delta & \zeta+\epsilon+\eta+\theta & \eta+\lambda+\iota+\kappa+\mu & \theta+\pi+\nu+\xi+o \\ \iota+\alpha+\beta+\gamma+\delta & \kappa+\zeta+\epsilon+\eta+\theta & \lambda+\iota+\kappa+\mu & \mu+\pi+\nu+\xi+o \\ \nu+\alpha+\beta+\gamma+\delta & \xi+\zeta+\epsilon+\eta+\theta & o+\lambda+\iota+\kappa+\mu & \pi+\nu+\xi+o \end{pmatrix}$$

Важливим уточненням є факт того, що завдяки структурі шифру Qalqan, атака на раунди $r-3$, $r-4$, ... призведе до нерозрізнюваних з точки зору диференціалів результатів у останньому раунді через накладання ключа $K^{(r-1)}$ та нелінійне перетворення S. Пошук кандидатів у байти ключа є значно складнішим, аніж у попередніх варіаціях через додавання значень веилкої кількості незалежних збоїв. Для спроби

відновлення байтів ключа можна віднімати елементи, які знаходяться у одному стовпчику. Це прибере з правої частини рівняння суму збоїв, отриману з діагонального елементу при останньому застосуванні лінійного перетворення L. Далі можна розглядати значення різниці байтів. як інше незалежне значення байту і перебирати його. Але при цьому у лівій частині рівняння виникне 4 елемента, які було розвернуто з останнього раунду. Далі необхідно перебирати значення відповідного ключа для цього стовпчика і шукати кандидатів у значення різниць, які задовольняють отримане збільшене рівняння. Однак більша складність перебору призводить до втрати ефективності обраного підходу до атаки.

Висновки до розділу 2

У цьому розділі було розглянуто застосування атак на Qalqan, в якому останній раунд має лише накладання ключа побітовим додаванням. Цей випадок є простішим за наступний, оскільки легше виконати обертання останнього раунду для побудови залежностей. Показано, що атаки з внесенням збою у раунди $r-1$ та $r-2$ дозволяють отримати інформацію про байти ключа, але атаку з внесенням збою у раунд $r-1$ доведеться провести 4 рази (наприклад на діагональних елементах) для отриманні інфорації про усі байти ключа $K^{(r)}$.

В результаті отримали, що атака з внесенням збою у раунд $r-2$ потребує меншу кількість збоїв, адже за один збій можна відновити усі байти ключа, або внести ще один для зменшення кількості переборів значення розв'язку рівнянь. При цьому, без уточнень кількість кандидатів складає 2^{64}, з уточеннями можна звести до єдиного значення.

Атака з внесенням збою у раунд $r-3$ з обраною конфігурацією не є ефективною, і потребує інших методів аналізу.

3 АТАКИ НА КЛЮЧ ПРОМІЖНИХ РАУНДІВ ШИФРУ QALQAN

У цьому розділі показано принцип застосування атак на узагальнений варіант проміжного раунду модифікованого шифру Qalqan. Як було зазначено у розділі 1.8, останнім раундом у моделі вважаємо накладання ключа $K^{(r)}$, застосування нелінійного перетворення S-блоку, лінійного перетворення L, і заключне накладання ключа $K^{(r+1)}$. Усі накладання відбуватимуться за допомогою побайтового додавання. Метою аналізу є отримання ключа $K^{(r+1)}$. Для цього розглядається внесення збоїв у раунди $r-1$, $r-2$ та $r-3$ і аналіз отриманих диференціалів.

3.1 Обернення лінійного перетворення L

На відміну від атаки на останній раунд шифру Qalqan, атака на проміжні раунди вимагає обернення останнього раунда, тому необхідно проаналізувати, як виглядає обернене до L лінійне перетворення.

Покажемо покроково, як виконується обернене перетворення. Нехай на виході буде отримано звичайний шифротекст $Y = (y_1 y_2 \dots y_{16})$ та збитий шифротекст $Y' = (y'_1 y'_2 \dots y'_{16})$. Проміжні матриці байтів при застосуванні оберненого лінійного перетворення L^{-1} матимуть наступний вигляд:

$$\begin{pmatrix} y_1 & y_2 & y_3 & y_4 \\ y_5 & y_6 & y_7 & y_8 \\ y_9 & y_{10} & y_{11} & y_{12} \\ y_{13} & y_{14} & y_{15} & y_{16} \end{pmatrix}$$

$$\begin{pmatrix} y_1 & y_2 - y_6 & y_3 - y_{11} & y_4 - y_{16} \\ y_5 - y_1 & y_6 & y_7 - y_{11} & y_8 - y_{16} \\ y_9 - y_1 & y_{10} - y_6 & y_{11} & y_{12} - y_{16} \\ y_{13} - y_1 & y_{14} - y_6 & y_{15} - y_{11} & y_{16} \end{pmatrix}$$

Фінальний вигляд покажемо у вигляді рівняння для кожного байту.

$$x_1 = y_1 + y_6 + y_{11} + y_{16} - y_2 - y_3 - y_4$$

$$x_2 = y_2 - y_6$$

$$x_3 = y_3 - y_{11}$$

$$x_4 = y_4 - y_{16}$$

$$x_5 = y_5 - y_1$$

$$x_6 = y_1 + y_6 + y_{11} + y_{16} - y_5 - y_7 - y_8$$

$$x_7 = y_7 - y_{11}$$

$$x_8 = y_8 - y_{16}$$

$$x_9 = y_9 - y_1$$

$$x_{10} = y_{10} - y_6$$

$$x_{11} = y_1 + y_6 + y_{11} + y_{16} - y_9 - y_{10} - y_{12}$$

$$x_{12} = y_{12} - y_{16}$$

$$x_{13} = y_{13} - y_1$$

$$x_{14} = y_{14} - y_6$$

$$x_{15} = y_{15} - y_{11}$$

$$x_{16} = y_1 + y_6 + y_{11} + y_{16} - y_{13} - y_{14} - y_{15}$$

3.2 Атака при внесенні збоїв на раунд $r-1$

Атака на раунд $r-1$ має за мету початковий аналіз залежностей і диференціалів а також оцінку можливості застосування атаки для безпосередньо відновлення інформації про ключ $K^{(r+1)}$.

Припустимо, що збій вноситься у x_1, де x_i — і-й байт шифротексту перед входом у раунд r. Тоді x'_1 - значення збитого байту, $x'_1 = x_1 + e$

Застосовується накладання ключа побайтовим додаванням, нелінійне перетворення S-блок, лінійне перетворення L і ще одне накладання ключа. Значення збою невідоме, збій поширився на увесь перший стовпчик. Позначимо збитий шифротекст як $y'_i = y_i + \alpha$. Позначимо байти ключа як k_i, $i = \overline{1,16}$.

При розвертання раунду назад, відбудеться віднімання ключа $K^{(r+1)}$ від шифротексту, застосування оберненого лінійного перетворення L^{-1} і нелінійного оберненого перетворення S^{-1}. За допомогою рівнянь покажемо залежність між значенням байтів збитого шифротексту, звичайного шифротексту, ключа $K^{(r+1)}$ та значенням збою перед початком останнього раунду r. Значення ключа $K^{(r)}$ скорочуються при відніманні обернених значень.

$$S^{-1}(y'_1 - k_1 + y'_6 - k_6 + y'_{11} - k_{11} + y'_{16} - k_{16} - y'_2 + k_2 - y'_3 + k_3 - y'_4 + k_4) -$$
$$- S^{-1}(y_1 - k_1 + y_6 - k_6 + y_{11} - k_{11} + y_{16} - k_{16} - y_2 + k_2 - y_3 + k_3 - y_4 + k_4) = e_1$$

$$S^{-1}(y'_2 - k_2 - y'_6 + k_6) - S^{-1}(y_2 - k_2 - y_6 + k_6) = 0$$

$$S^{-1}(y'_3 - k_3 - y'_{11} + k_{11}) - S^{-1}(y_3 - k_3 - y_{11} + k_{11}) = 0$$

$$S^{-1}(y'_4 - k_4 - y'_{16} + k_{16}) - S^{-1}(y_4 - k_4 - y_{16} + k_{16}) = 0$$

$$S^{-1}(y'_5 - k_5 - y'_1 + k_1) - S^{-1}(y_5 - k_5 - y_1 + k_1) = e_1$$

$$S^{-1}(y'_1 - k_1 + y'_6 - k_6 + y'_{11} - k_{11} + y'_{16} - k_{16} - y'_5 + k_5 - y'_7 + k_7 - y'_8 + k_8) -$$
$$- S^{-1}(y_1 - k_1 + y_6 - k_6 + y_{11} - k_{11} + y_{16} - k_{16} - y_5 + k_5 - y_7 + k_7 - y_8 + k_8) = 0$$

$$S^{-1}(y'_7 - k_7 - y'_{11} + k_{11}) - S^{-1}(y_7 - k_7 - y_{11} + k_{11}) = 0$$

$$S^{-1}(y'_8 - k_8 - y'_{16} + k_{16}) - S^{-1}(y_8 - k_8 - y_{16} + k_{16}) = 0$$

$$S^{-1}(y'_9 - k_9 - y'_1 + k_1) - S^{-1}(y_9 - k_9 - y_1 + k_1) = e_1$$

$$S^{-1}(y'_{10} - k_{10} - y'_6 + k_6) - S^{-1}(y_{10} - k_{10} - y_6 + k_6) = 0$$

$$S^{-1}(y'_1 - k_1 + y'_6 - k_6 + y'_{11} - k_{11} + y'_{16} - k_{16} - y'_9 + k_9 - y'_{10} + k_{10} - y'_{12} + k_{12}) -$$
$$- S^{-1}(y_1 - k_1 + y_6 - k_6 + y_{11} - k_{11} + y_{16} - k_{16} - y_9 + k_9 - y_{10} + k_{10} - y_{12} + k_{12}) = 0$$

$$S^{-1}(y'_{12} - k_{12} - y'_{16} + k_{16}) - S^{-1}(y_{12} - k_{12} - y_{16} + k_{16}) = 0$$

$$S^{-1}(y'_{13} - k_{13} - y'_1 + k_1) - S^{-1}(y_{13} - k_{13} - y_1 + k_1) = e_1$$

$$S^{-1}(y'_{14} - k_{14} - y'_6 + k_6) - S^{-1}(y_{14} - k_{14} - y_6 + k_6) = 0$$

$$S^{-1}(y'_{15} - k_{15} - y'_{11} + k_{11}) - S^{-1}(y_{15} - k_{15} - y_{11} + k_{11}) = 0$$

$$S^{-1}(y'_1 - k_1 + y'_6 - k_6 + y'_{11} - k_{11} + y'_{16} - k_{16} - y'_{13} + k_{13} - y'_{14} + k_{14} - y'_{15} + k_{15}) -$$
$$- S^{-1}(y_1 - k_1 + y_6 - k_6 + y_{11} - k_{11} + y_{16} - k_{16} - y_{13} + k_{13} - y_{14} + k_{14} - y_{15} + k_{15}) = 0$$

Необхідно зазначити, що байти шифротексту, різниця яких дає 0, замість значення збою, не несуть корисної для аналізу інформації і їх можна одразу відкинути. Для подальшого розв'язку необхідно перебрати усі можливі значення e_1; фіксуючи кожне значення, перебрати значення байту ключа k_1, при цьому роблячи перевірку, чи задовольняє значення різниць $k_1 - k_5$, $k_1 - k_9$ та $k_1 - k_{13}$ розв'язку рівняння. Таким чином знаходимо для конкретного значення e_1 та k_1 кандидатів у байти різниць. Далі, за допомогою повтору збою проводимо крос-валідацію і знижаємо кількість кандидатів. При цьому рівняння для діагональних елементів

можна розглядати за тим же принципом, але побудова співвідношень і перебір значень для них буде значно складніший.

Внесення збою у недіагональний елемент матриці байтів, як і у розділі 2.1 призведе до виникнення одного додаткового збитого елементу. Розв'язок рівнянь з урахуванням цього четвертого елементу системи дозволить зменшити кількість кандидатів відносно внесення збою у діагональний елемент. Але це уточнення буде компенсоване підвищеною складністю перебору через виникнення ще однієї різниці, яку необхідно врахувати при пошуку кандидатів.

Виділимо корисні для нас рівняння:

$$S^{-1}(y'_5 - k_5 - y'_1 + k_1) - S^{-1}(y_5 - k_5 - y_1 + k_1) = e_1 \tag{3.1}$$

$$S^{-1}(y'_9 - k_9 - y'_1 + k_1) - S^{-1}(y_9 - k_9 - y_1 + k_1) = e_1 \tag{3.2}$$

$$S^{-1}(y'_{13} - k_{13} - y'_1 + k_1) - S^{-1}(y_{13} - k_{13} - y_1 + k_1) = e_1 \tag{3.3}$$

Опишемо узагальнений алгоритм атаки з внесенням збоїв у раунд $r-1$:

1) Обираємо довільний відкритий текст X.

2) Зашифровуємо X за допомогою шифру Qalqan, отримуємо звичайний шифротекст Y.

3) Зашифровуємо X за допомогою шифру Qalqan, але у раунді $r-1$ вносимо збій у байт x_1, отримуємо збитий шифротекст Y'.

4) Аналітично розвертаємо раунд r назад зі значень Y та Y', і переходимо до входу у раунд r.

5) Перебираємо значення збою e_1.

6) Для кожного окремого значення збою, перебираємо усі можливі значення k_1.

7) Розв'язуємо рівняння (3.1) - (3.3) для різниць k_1-k_5, k_1-k_9, k_1-k_{13} і отримуємо кандидатів у значення різниць байтів.

8) Підставляємо обране значення k_1 у різницю і відновлюємо значення k_5, k_9, k_{13}.

9) Повторюємо кроки 3-7 тричі для уточнення обраних кандидатів

10) Повторюємо кроки 3-8 тричі для отримання значень у інших трьох ствопчиках, звідки дізнаємось значення k_i, де i - не діагональний елемент.

11) Брутфорсом відновлюємо значення діагональних байтів ключа.

12) Маючи усі значення байтів ключа, відновлюємо його.

У цьому алгоритмі можна пропустити крок 5, але від цього обмеження простору ключів буде значно зменшено.

Оціночні значення складності атаки зазначено на таблиці 3.1. Складність у диференційних атаках збоїв складається з кількості збоїв, які потрібно внести у виконання програми, результуючої кількості кандидатів у ключі і складність обчислення отриманих співвідношень (брутфорс рівнянь).

Таблиця 3.1 – Оціночні значення складності атаки з вносом збою у раунд $r - 1$

# збоїв	Складність розв'язку рівнянь	# кандидатів у ключі (в середньому)
4	$4 \cdot 2^{32}$	$12 \cdot 2^{10} \cdot 2^{32}$
8	$8 \cdot 2^{32}$	$\approx 2^{32}$
12	$12 \cdot 2^{32}$	2^{32}

Результуюча кількість кандидатів у ключі може бути значно зменшена, якщо розглядати додаткові залежності, які утворюються на діагональних позиціях матриці стану. Це підвищить складність перебору в певну константну кількість разів, але потенційно зменшить кількість кандидатів до 2^8.

3.3 Атака при внесенні збоїв на раунд $r - 2$

Конфігурація атаки буде схожею до описаної у розділі 3.2. Основною відмінністю від попереднього випадку є поширення диференціалу, яке повністю ідентично до описаного у розділі 2.2. За

допомогою рівнянь покажемо залежність між значенням байтів збитого шифротексту, звичайного шифротексту, ключа $K^{(r+1)}$ та значенням збою перед початком останнього раунду r. Аналогічно до 3.2, значення ключа $K^{(r)}$ скорочуються при відніманні обернених значень.

$$S^{-1}(y'_1 - k_1 + y'_6 - k_6 + y'_{11} - k_{11} + y'_{16} - k_{16} - y'_2 + k_2 - y'_3 + k_3 - y'_4 + k_4) -$$
$$- S^{-1}(y_1 - k_1 + y_6 - k_6 + y_{11} - k_{11} + y_{16} - k_{16} - y_2 + k_2 - y_3 + k_3 - y_4 + k_4) = e_1$$

$$S^{-1}(y'_2 - k_2 - y'_6 + k_6) - S^{-1}(y_2 - k_2 - y_6 + k_6) = e_2$$

$$S^{-1}(y'_3 - k_3 - y'_{11} + k_{11}) - S^{-1}(y_3 - k_3 - y_{11} + k_{11}) = e_3$$

$$S^{-1}(y'_4 - k_4 - y'_{16} + k_{16}) - S^{-1}(y_4 - k_4 - y_{16} + k_{16}) = e_4$$

$$S^{-1}(y'_5 - k_5 - y'_1 + k_1) - S^{-1}(y_5 - k_5 - y_1 + k_1) = e_1 + e_2$$

$$S^{-1}(y'_1 - k_1 + y'_6 - k_6 + y'_{11} - k_{11} + y'_{16} - k_{16} - y'_5 + k_5 - y'_7 + k_7 - y'_8 + k_8) -$$
$$- S^{-1}(y_1 - k_1 + y_6 - k_6 + y_{11} - k_{11} + y_{16} - k_{16} - y_5 + k_5 - y_7 + k_7 - y_8 + k_8) = e_2$$

$$S^{-1}(y'_7 - k_7 - y'_{11} + k_{11}) - S^{-1}(y_7 - k_7 - y_{11} + k_{11}) = e_3$$

$$S^{-1}(y'_8 - k_8 - y'_{16} + k_{16}) - S^{-1}(y_8 - k_8 - y_{16} + k_{16}) = e_4$$

$$S^{-1}(y'_9 - k_9 - y'_1 + k_1) - S^{-1}(y_9 - k_9 - y_1 + k_1) = e_1 + e_3$$

$$S^{-1}(y'_{10} - k_{10} - y'_6 + k_6) - S^{-1}(y_{10} - k_{10} - y_6 + k_6) = e_2$$

$$S^{-1}(y'_1 - k_1 + y'_6 - k_6 + y'_{11} - k_{11} + y'_{16} - k_{16} - y'_9 + k_9 - y'_{10} + k_{10} - y'_{12} + k_{12}) -$$
$$- S^{-1}(y_1 - k_1 + y_6 - k_6 + y_{11} - k_{11} + y_{16} - k_{16} - y_9 + k_9 - y_{10} + k_{10} - y_{12} + k_{12}) = e_2$$

$$S^{-1}(y'_{12} - k_{12} - y'_{16} + k_{16}) - S^{-1}(y_{12} - k_{12} - y_{16} + k_{16}) = e_3$$

$$S^{-1}(y'_{13} - k_{13} - y'_1 + k_1) - S^{-1}(y_{13} - k_{13} - y_1 + k_1) = e_1 + e_4$$

$$S^{-1}(y'_{14} - k_{14} - y'_6 + k_6) - S^{-1}(y_{14} - k_{14} - y_6 + k_6) = e_2$$

$$S^{-1}(y'_{15} - k_{15} - y'_{11} + k_{11}) - S^{-1}(y_{15} - k_{15} - y_{11} + k_{11}) = e_3$$

$$S^{-1}(y'_1 - k_1 + y'_6 - k_6 + y'_{11} - k_{11} + y'_{16} - k_{16} - y'_{13} + k_{13} - y'_{14} + k_{14} - y'_{15} + k_{15}) -$$
$$- S^{-1}(y_1 - k_1 + y_6 - k_6 + y_{11} - k_{11} + y_{16} - k_{16} - y_{13} + k_{13} - y_{14} + k_{14} - y_{15} + k_{15}) = e_4$$

Можна побачити, що усі байти містять ненульове значення у правій частині рівняння. Це робить дану атаку більш ефективною з точки зору кількості збоїв, які необхідно вставити у програму для відновлення усіх байтів ключа. Оскільки e_1 додається до інших збоїв, рекомендованою стратегією є внесення нового збою у інший стовпчик для пошуку кандидатів у різницях $k_1 - k_5$, $k_1 - k_9$, $k_1 - k_{13}$. Рівняння для діагональних елементів є складнішими для утворення співвідношені і перебору.

Для розв'язку розбиваємо рівняння на окремі системи з однаковим значенням збою у правій частині. Для кожної утвореної системи перебираємо значення збою, фіксуємо його, перебираємо значення байтів ключа k_6 для e_2, k_{11} для e_3, k_{16} для e_4 і шукаємо значення різниць, для яких виконуються рівняння. Таким чином одразу в трьох блоках паралельно можна відновити 9 байтів ключа.

Опишемо узагальнений алгоритм атаки з внесенням збоїв у раунд $r - 1$:

1) Обираємо довільний відкритий текст X.

2) Зашифровуємо X за допомогою шифру Qalqan, отримуємо звичайний шифротекст Y.

3) Зашифровуємо X за допомогою шифру Qalqan, але у раунді $r - 1$ вносимо збій у байт x_1, отримуємо збитий шифротекст Y'.

4) Аналітично розвертаємо раунд r назад зі значень Y та Y', і переходимо до входу у раунд r.

5) Перебираємо значення збою e_2.

6) Для кожного окремого значення збою, перебираємо усі можливі значення k_6.

7) Розв'язуємо рівняння (3.1) - (3.3) для різниць $k_6 - k_2$, $k_6 - k_{10}$,

$k_6 - k_{14}$ і отримуємо кандидатів у значення різниць байтів.

8) Підставляємо обране значення k_6 у різницю і відновлюємо значення k_2, k_{10}, k_{14}.

9) Повторюємо кроки 3-7 тричі для уточнення обраних кандидатів

10) Повторюємо кроки 5-9 для збоїв e_3, e_4.

11) Повторюємо кроки 3-9, але збиваємо інший діагональний елемент, який не знаходиться на першому рядку, відновлюємо значення k_5, k_9, k_{13}

12) Брутфорсом відновлюємо значення діагональних байтів ключа.

13) Маючи усі значення байтів ключа, відновлюємо його.

У цьому алгоритмі можна пропустити крок 5, але від цього обмеження простору ключів буде значно зменшено.

Оціночні значення складності атаки зазначено на таблиці 3.2. Складність у диференційних атаках збоїв складається з кількості збоїв, які потрібно внести у виконання програми, результуючої кількості кандидатів у ключі і складність обчислення отриманих співвідношень (брутфорс рівнянь).

Таблиця 3.2 – Оціночні значення складності атаки з вносом збою у раунд $r - 2$

# збоїв	Складність розв'язку рівнянь	# кандидатів у ключі (в середньому)
2	$4 \cdot 2^{32}$	$12 \cdot 2^{10} \cdot 2^{32}$
4	$8 \cdot 2^{32}$	$\approx 2^{32}$
6	$12 \cdot 2^{32}$	2^{32}

Як і у попередньому випадку, кількість кандидатів може бути зменшена до 2^8 при обчисленні додаткових залежностей між діагональними байтами ключів.

3.4 Аналіз диференціалів при внесенні збоїв у раунд $r-3$

Аналогічно до розділу 2.3, випадки атаки на діагональний і не діагональний елемент нерозрізнювані з точки зору диференціалів, тому окремо розглядати їх немає необхідності.

$$S^{-1}(y'_1 - k_1 + y'_6 - k_6 + y'_{11} - k_{11} + y'_{16} - k_{16} - y'_2 + k_2 - y'_3 + k_3 - y'_4 + k_4) -$$
$$-S^{-1}(y_1 - k_1 + y_6 - k_6 + y_{11} - k_{11} + y_{16} - k_{16} - y_2 + k_2 - y_3 + k_3 - y_4 + k_4) = e_1 + e_2 + e_3 + e_4$$

$$S^{-1}(y'_2 - k_2 - y'_6 + k_6) - S^{-1}(y_2 - k_2 - y_6 + k_6) = e_2 + e_5 + e_6 + e_7 + e_8$$

$$S^{-1}(y'_3 - k_3 - y'_{11} + k_{11}) - S^{-1}(y_3 - k_3 - y_{11} + k_{11}) = e_3 + e_9 + e_{10} + e_{11} + e_{12}$$

$$S^{-1}(y'_4 - k_4 - y'_{16} + k_{16}) - S^{-1}(y_4 - k_4 - y_{16} + k_{16}) = e_4 + e_{13} + e_{14} + e_{15} + e_{16}$$

$$S^{-1}(y'_5 - k_5 - y'_1 + k_1) - S^{-1}(y_5 - k_5 - y_1 + k_1) = e_5 + e_1 + e_2 + e_3 + e_4$$

$$S^{-1}(y'_1 - k_1 + y'_6 - k_6 + y'_{11} - k_{11} + y'_{16} - k_{16} - y'_5 + k_5 - y'_7 + k_7 - y'_8 + k_8) -$$
$$-S^{-1}(y_1 - k_1 + y_6 - k_6 + y_{11} - k_{11} + y_{16} - k_{16} - y_5 + k_5 - y_7 + k_7 - y_8 + k_8) = e_5 + e_6 + e_7 + e_8$$

$$S^{-1}(y'_7 - k_7 - y'_{11} + k_{11}) - S^{-1}(y_7 - k_7 - y_{11} + k_{11}) = e_7 + e_9 + e_{10} + e_{11} + e_{12}$$

$$S^{-1}(y'_8 - k_8 - y'_{16} + k_{16}) - S^{-1}(y_8 - k_8 - y_{16} + k_{16}) = e_8 + e_{13} + e_{14} + e_{15} + e_{16}$$

$$S^{-1}(y'_9 - k_9 - y'_1 + k_1) - S^{-1}(y_9 - k_9 - y_1 + k_1) = e_9 + e_1 + e_2 + e_3 + e_4$$

$$S^{-1}(y'_{10} - k_{10} - y'_6 + k_6) - S^{-1}(y_{10} - k_{10} - y_6 + k_6) = e_{10} + e_5 + e_6 + e_7 + e_8$$

$$S^{-1}(y'_1 - k_1 + y'_6 - k_6 + y'_{11} - k_{11} + y'_{16} - k_{16} - y'_9 + k_9 - y'_{10} + k_{10} - y'_{12} + k_{12}) -$$
$$- S^{-1}(y_1 - k_1 + y_6 - k_6 + y_{11} - k_{11} + y_{16} - k_{16} - y_9 + k_9 - y_{10} + k_{10} - y_{12} + k_{12}) =$$
$$= e_9 + e_{10} + e_{11} + e_{12}$$

$$S^{-1}(y'_{12} - k_{12} - y'_{16} + k_{16}) - S^{-1}(y_{12} - k_{12} - y_{16} + k_{16}) = e_{12} + e_{13} + e_{14} + e_{15} + e_{16}$$

$$S^{-1}(y'_{13} - k_{13} - y'_1 + k_1) - S^{-1}(y_{13} - k_{13} - y_1 + k_1) = e_{13} + e_1 + e_2 + e_3 + e_4$$

$$S^{-1}(y'_{14} - k_{14} - y'_6 + k_6) - S^{-1}(y_{14} - k_{14} - y_6 + k_6) = e_{14} + e_5 + e_6 + e_7 + e_8$$

$$S^{-1}(y'_{15} - k_{15} - y'_{11} + k_{11}) - S^{-1}(y_{15} - k_{15} - y_{11} + k_{11}) = e_{15} + e_9 + e_{10} + e_{11} + e_{12}$$

$$\begin{aligned} S^{-1}(y'_1 - k_1 + y'_6 - k_6 + y'_{11} - k_{11} + y'_{16} - k_{16} - y'_{13} + k_{13} - y'_{14} + k_{14} - y'_{15} + k_{15}) - \\ - S^{-1}(y_1 - k_1 + y_6 - k_6 + y_{11} - k_{11} + y_{16} - k_{16} - y_{13} + k_{13} - y_{14} + k_{14} - y_{15} + k_{15}) = \\ = e_{13} + e_{14} + e_{15} + e_{16} \end{aligned}$$

Важливо зазначити, що, аналогічно до розділу 2.3, завдяки структурі шифру Qalqan, внесення збоїв у раунди $r-3$, $r-4$ і старших призведе до нерозрізнюваних з точки зору диференціалів результатів на вході у останній раунд r через накладання ключа $K^{(r-1)}$ та нелінійне перетворення S. Пошук кандидатів у байти ключа є значно складнішим, аніж у попередніх варіаціях через додавання значень збоїв. Для спроби відновлення байтів ключа можна віднімати елементи, які знаходяться у одному стовпчику. Це прибере з правої частини рівняння суму збоїв, отриману з діагонального елементу при останньому застосуванні лінійного перетворення L. Далі можна розглядати значення різниці байтів. як інше незалежне значення байту і перебирати його. Але при цьому у лівій частині рівняння виникне 4 елемента, які було розвернуто з останнього раунду. Далі необхідно перебирати значення відповідного ключа для цього стовпчика і шукати кандидатів у значення різниць, які задовольняють отримане збільшене рівняння. Однак більша складність перебору призводить до втрати ефективності обраного підходу до атаки.

Висновки до розділу 3

У цьому розділі було розглянуто атаку на узагальнений вигляд проміжного раунду модифікованого шифру Qalqan. Показано, що

принцип атак мало відрізняється від атак на узагальнений вигляд останнього раунду. Можна побачити, що, аналогічно до 2, при використанні атаки з внесенням збою у раунд $r-1$ потребується більша кількість збоїв для отримання інформації про усі байти раундового ключа. При цьому, пошук кандидатів у байти ключа для атаки з внесенням збою у раунд $r-3$ перестає бути ефективним через велику кількість переборів для розв'язку рівнянь.

В результаті отримали, що атака з внесенням збою у раунд $r-2$ потребує меншу кількість збоїв, адже за один збій можна відновити усі байти ключа, або внести ще один для зменшення кількості переборів значення розв'язку рівнянь. При цьому, без уточнень кількість кандидатів складає приблизно 2^{46}, з уточненнями можна звести до 2^{32}.

ВИСНОВКИ

У даній роботі розглядались диференціальні атаки збоїв на блокові шифри. Було розглянуто існуючі підходи та методи до побудови диференціальних атак збоїв; проаналізовано структуру та особливості шифру Qalqan. Показано, що за рахунок слабкого лавинного ефекту лінійного перетворення L можна побудувати ефективні атаки збоїв при внесенні збою за декілька раундів до фінального.

За пропоновано атаки, які здатні ефективно відновити значення байтів ключа останнього раунду, яка враховує операцію побітового додавання $\oplus$ у останньому накладанні ключа. Запропоновано атаки, які здатні ефективно відновити значення байтів ключа проміжних раундів, які враховують операцію побайтового додавання $+$ при накладанні ключа у цих раундах.

Отримано оцінки кількості збоїв, кандидатів у байти ключа і кількості переборів для розв'язку рівнянь. Для атаки на останній раунд з використанням збою у раунді $r-1$ необхідно 4 збоя, 8 чи 12 для уточнень, отримується кількість кандидатів 2^{64} (що можна за допомогою уточнень звести до єдиного кандидату) і необхідно приблизно 2^{20} застосувань перебору (не більше 2^{24} для уточень). Для атаки на останній раунд з використанням збою у раунді $r-2$ необхідно 2 збоя, 4 чи 6 для уточнень, отримується кількість кандидатів 2^{64} (що можна за допомогою уточнень звести до єдиного кандидату) і необхідно приблизно 2^{20} застосувань перебору (не більше 2^{24} для уточень).

Для атаки на проміжні раунди з використанням збою у раунді $r-1$ необхідно 4 збоя, 8 чи 12 для уточнень, отримується кількість кандидатів 2^{46} (що можна за допомогою уточнень звести до 2^{32}) і необхідно приблизно 2^{34} застосувань перебору (не більше 2^{36} для уточень). Для атаки на проміжні раунд з використанням збою у раунді $r-2$ необхідно 4 збоя, 8 чи 12 для уточнень, отримується кількість кандидатів 2^{46} (що

можна за допомогою уточнень звести до 2^{32}) і необхідно приблизно 2^{34} застосувань перебору (не більше 2^{36} для уточень). При цьому, результуючу кількість кандидатів у ключі можна значно зменшити, якщо розглядати додаткові залежності, які утворюються на діагональних позиціях матриці стану. Це підвищить складність перебору в певну константну кількість разів (що може призвести до зниження ефективності, але не до унеможливлення пошуку), але потенційно зменшить кількість кандидатів до 2^8.

Можливими напрямками подальшого дослідження є уточнення отриманих експериментальних результатів атаки і перевірка їх з різними конфігураціями ключів, побудова і застосування атаки з вносом збою у раунд $r - 3$, чи старших раундів, покращення запропонованих атак по кількості внесених збоїв, складності брутфорсу і зменшення кількості кандидатів, перевірка застосовності запропонованих атак до інших шифрів з байт-орієнтованою архітектурою, які використовують байтові операції, або інші види операцій за модулем.

ПЕРЕЛІК ПОСИЛАНЬ

[1] A. Baksi та ін. *BAKSHEESH: Similar Yet Different From GIFT*. Cryptology ePrint Archive, Paper 2023/750. https://eprint.iacr.org/2023/750. 2023. URL: https://eprint.iacr.org/2023/750.

[2] A. Baksi та ін. *DEFAULT: Cipher Level Resistance Against Differential Fault Attack*. Cryptology ePrint Archive, Paper 2021/712. https://eprint.iacr.org/2021/712. 2021. URL: https://eprint.iacr.org/2021/712.

[3] S. Banik та ін. *GIFT: A Small Present*. Cryptology ePrint Archive, Paper 2017/622. https://eprint.iacr.org/2017/622. 2017. URL: https://eprint.iacr.org/2017/622.

[4] L. Gorlov та ін. «Алгоритм шифрования Qalqan». В: *Proceedings of VI International Scientific Conference "Computer Science and Applied Mathematics"* (29 жовт. 2021), 458–463. URL: https://conf.iict.kz/wp-content/uploads/2021/11/6th_csam.29.09-02.10.21_sbornik.pdf.

[5] L. Hemme. «A Differential Fault Attack Against Early Rounds of (Triple-)DES». В: *Cryptographic Hardware and Embedded Systems - CHES 2004*. За ред. M. Joye та J.J. Quisquater. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, с. 254—267. ISBN: 978-3-540-28632-5.

[6] A. Jana, A. Kumar Kundu та Goutam P. *SDFA: Statistical-Differential Fault Attack on Linear Structured SBox-Based Ciphers*. Cryptology ePrint Archive, Paper 2023/1018. https://eprint.iacr.org/2023/1018. 2023. URL: https://eprint.iacr.org/2023/1018.

[7] M. Joye та M. Tunstall. *Differential Fault Analysis*. 2023. URL: https://marcjoye.github.io/papers/JT23dfa.pdf.

[8] M. Nageler, C. Dobraunig та M. Eichlseder. *Information-Combining Differential Fault Attacks on DEFAULT*. Cryptology ePrint Archive, Paper 2021/1374. https : / / eprint . iacr . org / 2021 / 1374. 2021. URL: https://eprint.iacr.org/2021/1374.

[9] G. Piret та J. J. Quisquater. «A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad». В: *CHES 2003* (2003), с. 77—88.

[10] M. Tunstall, D. Mukhopadhyay та S. Ali. «Differential Fault Analysis of the Advanced Encryption Standard Using a Single Fault». В: *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication*. За ред. C. A. Ardagna та J. Zhou. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, с. 224—233. ISBN: 978-3-642-21040-2.

[11] S. Yakovliev та M. Stolovych. «On the Security of Qalqan Cipher against Differential Cryptanalysis». В: *Theoretical And Applied Cybersecurity* 4.1 (2022). URL: https://doi.org/10.20535/tacs.2664-29132022.1.274112.

[12] М. А. Недождій та С. В. Яковлєв. «Побудова диференціальної атаки збоїв на модифікований шифр Qalqan». В: *Кіберборотьба: розвідка, захист та протидія: тези доповідей II Міжнародної науково-практичної конференції*. Київ: Військовий інститут телекомунікацій та інформатизації імені Героїв Крут, 2024, с. 37—38. URL: https://cyberwarfare.viti.edu.ua/.

[13] М. А. Недождій та С. В. Яковлєв. «Побудова диференціальної атаки збоїв на модифікований шифр Qalqan». В: *XXII Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» (13-17 травня 2024 р., м. Київ, Україна) : матеріали конференції*. Київ: КПІ ім. Ігоря Сікорського, Видавництво «Політехніка», 2024, с. 228—230. ISBN: 978-966-990-053-1.

[14] С. Яковлєв. «Атаки збоїв на шифр ДСТУ ГОСТ 28147:2009». В: *Інформаційна безпека людини, суспільства, держави* №2(18) (2015), с. 124—136.