НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра математичних методів захисту інформації

«До захисту допущено»
В.о. завідувача кафедри
_____ Михайло САВЧУК
«__» _____ 2021 р.

Дипломна робота

на здобуття ступеня бакалавра

зі спеціальності: 113 Прикладна математика
на тему: **«Криптоаналіз малоресурсного симетричного блокового шифру „Кипарис“»**

Виконав: студент 4 курсу, групи ФІ-73
Звичайна Анастасія Олександрівна

Керівник: ст.викладач, к.ф.-м.н. Фесенко А.В. _____

Консультант: _ _____

Рецензент: доцент, к.ф.-м.н. Южакова Г.О. _____

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

Студент _____

Київ — 2021

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра математичних методів захисту інформації

Рівень вищої освіти — перший (бакалаврський)
Спеціальність (освітня програма) — 113 Прикладна математика,
ОПП «Математичні методи криптографічного захисту інформації»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Михайло САВЧУК
«__» _____ 2021 р.

ЗАВДАННЯ
на дипломну роботу

Студент: Звичайна Анастасія Олександрівна

1. Тема роботи: «*Криптоаналіз малоресурсного симетричного блокового шифру „Кипарис“*»,

керівник: ст.викладач, к.ф.-м.н. Фесенко А.В.,

затверджені наказом по університету №__ від «__» _____ 2021 р.

2. Термін подання студентом роботи: 4 червня 2021 р.

3. Вихідні дані до роботи: *специфікація шифру «Кипарис», статті з ротаційного та квантового криптоаналізів.*

4. Зміст роботи: *огляд методів криптоаналізу ARX-криптосистем; квантові атаки на фейстель-подібні схеми; оцінка стійкості шифру «Кипарис» до квантових та ротаційних атак.*

5. Перелік ілюстративного матеріалу: *презентація доповіді.*

6. Дата видачі завдання: 10 вересня 2020 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання	Примітка
1	Обговорення можливих тем роботи із науковим керівником	Вересень 2020 р.	Виконано
2	Огляд методів криптоаналізу, зокрема R- та RX-криптоаналізів, а також їх застосунків до наявних ARX-криптопримітивів. Перше знайомство з шифром «Кипарис». Аналіз робіт з диференціального криптоаналізу	Жовтень-листопад 2020 р.	Виконано
3	Огляд наявних досліджень стійкості та продуктивності шифру «Кипарис». Проведення ротаційного криптоаналізу «Кипарис»	Грудень 2020 р.	Виконано
4	Аналіз робіт з диференціального, лінійного та ротаційного криптоаналізів	Лютий-березень 2021 р.	Виконано
5	Опрацювання робіт з квантового криптоаналізу фейстель-подібних схем. Проведення поліноміальної атаки на трираундову схему шифрування «Кипарис». Дослідження стійкості узагальненої схеми Фейстеля типу III у квантовій моделі обчислень	Квітень 2021 р.	Виконано
6	Проведення субекспоненційної атаки на п'ять раундів шифру «Кипарис». Оцінки складності проведення квантових атак на всю схему шифрування «Кипарис». Оформлення пояснювальної записки до роботи	Травень 2021 р.	Виконано

Студент _____ Звичайна А.О.

Керівник _____ Фесенко А.В.

РЕФЕРАТ

Кваліфікаційна робота містить: 92 сторінки, 10 рисунків, 3 таблиці, 77 джерел.

У роботі вперше проводиться квантовий криптоаналіз малоресурсного симетричного блокового шифру «Кипарис», за результатами якого на його трираундову схему шифрування побудована поліноміальна атака розрізнення випадкового тексту від шифротексту, а також субекспоненційна атака відновлення ключів на четвертому та п'ятому раундах. Окрім цього, отримані оцінки складності проведення атак для повної кількості раундів у квантовій моделі обчислень. Також доведена стійкість до ротаційного криптоаналізу версії «Кипарис-256» та отримана ймовірність ротації для «Кипарис-512», яка майже не відрізняється від ймовірності випадкової перестановки. У роботі доведена вразливість узагальненої схеми Фейстеля типу III у квантовій моделі обчислень, де кількість раундів збігається з кількістю частин, на які розбивається відкритий текст на вході у схему шифрування.

Метою роботи є дослідження стійкості малоресурсного симетричного блокового шифру «Кипарис» до квантового та ротаційного криптоаналізів.

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

Предметом дослідження є стійкість до квантового та ротаційного криптоаналізів малоресурсного симетричного блокового шифру «Кипарис».

МАЛОРЕСУРСНА КРИПТОГРАФІЯ, СИМЕТРИЧНА КРИПТОГРАФІЯ, СХЕМИ ФЕЙСТЕЛЯ, ARX-ШИФРИ, «КИПАРИС», КВАНТОВИЙ КРИПТОАНАЛІЗ, ЗАДАЧА САЙМОНА, ЗАДАЧА ГРОВЕРА, ARX-АНАЛІЗ, РОТАЦІЙНИЙ КРИПТОАНАЛІЗ

ABSTRACT

The thesis contains: 92 pages, 10 figures, 3 tables, 77 sources.

In this thesis, for the first time, a quantum cryptanalysis of the lightweight symmetric block cipher «Cypress» is introduced. According to the results of which a polynomial attack of distinguishing a random text from a ciphertext was carried out againts its three-round encryption scheme, and a subexponential key-recovery attack againts its fourth and fifth rounds. Furthermore in this thesis were obtained estimates the complexity of attacks for the total number of rounds in the quantum adversary model. Proven resistance to rotational attacks for version «Cypress-256». The obtained probability of rotational cryptanalysis against «Cypress-512» is close to the probability of a random permutation. It is also proven that the type III Generalized Feistel Scheme is not immune against quantum computing attacks if the number of rounds coincides with the number of parts into which the plaintext is divided at the entrance to the encryption scheme.

The purpose of the thesis is investigate the resistance of the lightweight symmetric block cipher «Cypress» to quantum and rotational cryptanalysis.

The research object is information processes in information security systems.

The research subject is resistance to quantum and rotational cryptanalysis of lightweight symmetric block cipher «Cypress».

LIGHTWEIGHT CRYPTOGRAPHY, SYMMETRIC CRYPTOGRAPHY, FEISTEL CIPHERS, ARX-CIPHERS, «CYPRESS», QUANTUM CRYPTANALYSIS, SIMON'S PROBLEM, GROVER'S PROBLEM, ARX-ANALYSIS, ROTATIONAL CRYPTANALYSIS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

$+$ — звичайне шкільне додавання

$x = (x_{n-1},...,x_1,x_0)$ — n-бітний двійковий вектор; x_0 - молодший значущий біт

$\boxplus$ — додавання за модулем

$\lll$ — циклічний зсув вліво

$\ggg$ — циклічний зсув вправо

$\oplus$ — виключна диз'юнкція (XOR)

$\overleftarrow{x}$ — $x \lll 1$

$\mathbb{F}_{2^n}$ — поле Галуа з 2^n елементами

$\mathbb{M}_{\text{mxm}}(\mathbb{F}_{2^n})$ — набір матриць розміру $m \times m$, елементи яких належать $\mathbb{F}_{2^n}$

$x \parallel y$ — конкатенація двійкових векторів x та y

$x \cdot y$ — скалярний добуток векторів x та y з дійсними коефіцієнтами (звичайне шкільне множення)

$O(\cdot)$ — «О» велике за нотацією Ландау

$\langle \cdot |$ — бра-вектор за нотацією Дірака

$| \cdot \rangle$ — кет-вектор за нотацією Дірака

$\langle x|y\rangle$ — скалярний добуток векторів $\langle x|$ та $|y\rangle$

$A \otimes B$ — кронекеровий добуток матриць A та B

$dim(u_1,...,u_l)$ — розмірність лінійної оболонки, що утворена векторами $u_1,...,u_l$

$|x|$ — вага Хемінга вектора x

$\binom{n}{k}$ — біноміальний коефіцієнт з n по k

$SHL(x)$ — нециклічний зсув x вліво на один біт

$1_{\{Z\}}$ — індикатор, який дорівнює 1 якщо Z має місце на існування

$(I \oplus SHL)(x)$ — $x \oplus SHL(x)$

$x|y$ — побітове АБО між x та y

$R(x)$ — $n - \gamma$ наймолодших значущих бітів n-бітного вектора x (γ деяке число, що менше за n)

$L'(x)$ — $n - \gamma$ найстарших значущих бітів n-бітного вектора x (γ деяке число, що менше за n)

$\boxplus_{i=0}^{k} x_i$ — додавання за модулем від 0 до k двійкових векторів x_i

$\vee$ — логічне АБО

$\sum_{i=0}^{k} x_i$ — сума від 0 до k для чисел x_i

$h^2(\cdot)$ — двократне застосування функції h, тобто $h(h(\cdot))$

ВСТУП

Актуальність дослідження зумовлена збільшенням кількості малоресурсних пристроїв у світі внаслідок поширеного застосування мобільних телефонів, смарткарток, RFID-міток, сенсорних мереж та стрімкого розвитку Інтернету речей, основною задачею яких є збір та передача інформації. ARX-шифри найкраще справляються з захистом інформації на таких пристроях, але питання безпеки є відкритим, тому останніми роками все більше і більше досліджень присвячені саме криптоаналізу ARX-шифрів. В Україні таким шифром є «Кипарис», який поки що не є стандартизованим, але має всі шанси на це.

Окрім цього, актуальність зумовлена стрімким розвитком квантових комп'ютерів, які, завдяки властивостям квантової фізики, мають безумовну перевагу у швидкості обчислень. Так, якщо у 1982 році, коли Річард Фейнман висловив ідею щодо створення такого комп'ютера, ніхто не вірив у можливу його реалізацію, то сьогодні вже ніхто не сумнівається у можливості створення потужного квантового комп'ютера та його загрозі для сучасних криптопримітивів, особливо малоресурсних.

Метою роботи є дослідження стійкості малоресурсного симетричного блокового шифру «Кипарис» до квантового та ротаційного криптоаналізів. Для досягнення мети необхідно виконати такі завдання:

1) провести огляд методів криптоаналізу, зокрема ротаційного та квантового, а також їх застосунків до наявних ARX-схем;

2) дослідити стійкість класичної трираундової схеми Фейстеля та побудувати атаку на три раунди схеми шифрування «Кипарис» у квантовій моделі обчислень;

3) побудувати атаку відновлення ключів на усічену версію шифру «Кипарис» та дослідити стійкість повної версії шифру «Кипарис» у квантовій моделі обчислень;

4) дослідити стійкість шифру «Кипарис» до ротаційного

криптоаналізу;

5) дослідити стійкість узагальненої схеми Фейстеля типу III у квантовій моделі обчислень.

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

Предметом дослідження є стійкість до квантового та ротаційного криптоаналізів малоресурсного симетричного блокового шифру «Кипарис».

При розв'язанні поставлених завдань використовувались такі **методи дослідження**: теорії ймовірності, комбінаторного аналізу, квантових обчислень, теорії складності алгоритмів, лінійної та абстрактної алгебри.

Наукова новизна дослідження полягає у проведенні квантового криптоаналізу малоресурсного симетричного блокового шифру «Кипарис»: побудові поліноміальної атаки на його трираундову схему шифрування, побудові субекспоненційної атаки на п'ятираундову схему, а також загальної оцінки його стійкості до квантового криптоаналізу. Окрім цього, вперше доведена стійкість версії «Кипарис-256» до ротаційного криптоаналізу, а для версії «Кипарис-512» отримана ймовірність ротації є близькою до ймовірності випадкової перестановки. Також вперше доведена вразливість узагальненої схеми Фейстеля типу III у квантовій моделі обчислень, де кількість раундів збігається з кількістю частин, на які розбивається відкритий текст на вході у схему шифрування.

Практичне значення одержаних результатів полягає у можливості використання малоресурсного симетричного блокового шифру «Кипарис» для захисту інформації у постквантовий період, а також у неможливості застосування криптопримітивів, в основі яких лежить узагальнена схема Фейстеля типу III, де кількість раундів збігається з кількістю частин, на які розбивається відкритий текст на вході у схему шифрування.

1 ARX-ШИФРИ ТА МЕТОДИ ЇХ КРИПТОАНАЛІЗУ

У цьому розділі обґрунтовується необхідність розробки ARX-примітивів, розглядаються деякі їх приклади, зокрема будова малоресурсного симетричного блокового шифру «Кипарис». Описуються наявні методи криптоаналізу ARX-шифрів, починаючи з огляду диференціального та лінійного криптоаналізів, які є класичними методами, закінчуючи ротаційним та квантовим криптоаналізами, а також комбінаціями різних методів.

1.1 Необхідність ARX-примітивів та їх приклади

Зараз відбувається стрімкий розвиток та впровадження у життя малоресурсних пристроїв, а також технологій, що об'єднують їх у єдину мережу, серед яких бездротові сенсорні мережі (WSN) та Інтернет речей (IoT) [1]. Найяскравішими прикладами використання WSN та IoT є такі системи як розумний дім, розумне місто, розумний транспорт. Однак мало хто знає, що понад 40% усього Інтернету речей спрямовано на медичні потреби [2], адже розумне застосування IT технологій у сфері охорони здоров'я може пришвидшити одужання пацієнтів, значно покращити їх стан, а також попередити багато захворювань. Більше не треба буде розшифровувати рецепти лікарів, шукати старі медичні картки. Все необхідне може знаходитись у деякій базі даних, яка оновлюється у режимі реального часу. Система неперервного моніторингу рівня глюкози (CGM) є сьогодні однією з найактуальніших систем охорони здоров'я, адже дає змогу лікарям і пацієнтам слідкувати за своїм станом кожної секунди [3]. Ще одним прикладом є співпраця компанії Intel з Орегонським університетом здоров'я та науки, результатом якої є проєкт, що допомагає лікарям досліджувати рак. Існують й інші

приклади у сфері медицини, з якими можна ознайомитись у статті [2]. Очевидно, що основною задачею таких пристроїв та систем є збір та передача інформації, яка у деяких випадках може бути дуже конфіденційною, тому виникає необхідність розробки стійких алгоритмів, враховуючи обмежену обчислювану потужність та пам'ять зазначених пристроїв. Така потреба стала поштовхом для появи «легкої криптографії», розробники якої повинні знайти компроміс між безпекою, витратами та ефективністю реалізації криптопримітивів. Зазвичай лише дві з трьох цілей можна досягнути, тому, маючи недорогі пристрої, гострим стає питання безпеки, адже швидкістю обчислень не можна нехтувати [4]. Відомо, що криптографічні примітиви, що використовують лише операції додавання за модулем (нелінійна операція), виключної диз'юнкції та циклічного зсуву працюють найшвидше, а шифри/схеми, що вони реалізують, називають ARX-шифрами/схемами, що є абревіатурою трьох операцій (англ. modular addition, cyclic rotation та exclusive-or (XOR)). Перший такий шифр розробили у 1987 році (шифр FEAL [5]), а саме поняття ARX-шифру/схеми з'явилось лише у 2009 році у роботі Ральф-Філіпа Вайнмана (англ. Ralf-Philipp Weinmann) [6], після якої кількість таких схем значно виросла. Найвідомішими ARX-схемами є геш-функції BLAKE2 (досліджувалась саме функція перестановки, США) [7], Skein (відповідно і блоковий шифр Threefish, США) [8], блокові шифри SPECK (США) [9], SPARX [10], LEA (Південна Корея) [11], потокові шифри Salsa20 [12] та ChaCha [13] (досліджувались саме функції перестановки), алгоритм із виробленням імітоставки Chaskey [14]. В Україні одним з представників ARX-шифрів є «Кипарис» [15], який поки що не є стандартизованим, але має всі шанси на це. Розглянемо детальніше деякі з таких схем.

BLAKE2

У 2007 році NIST (Національний інститут стандартів і технології) оголосив конкурс «SHA-3», суть якого полягала у тому, щоб прийняти нову криптографічну геш-функцію у якості стандарту. У 2011 році

оголосили фіналістів конкурсу, серед яких була геш-функція BLAKE, а у 2012 році — результати конкурсу. Хоч BLAKE і не стала переможцем, але на її основі створили геш-функцію BLAKE2 [7], функцію перестановки якої розглянемо. Нехай P — перестановка у BLAKE2 (див. рис. 1.1), на вхід до якої подається шістнадцять 64-бітових векторів-станів v_i та повідомлення, що розбито на шістнадцять 64-бітових векторів m_i. P має 12 раундів, кожний з яких 8 разів використовує функцію

$$G_i(a,b,c,d) = G(a,b,c,d,m_{f(i)},m_{g(i)}),$$

де $f(i)$ та $g(i)$ реалізують перестановку на шістнадцяти 64-бітових векторах m_i. Функція G працює таким чином:

1. $a = a \boxplus b \boxplus m_{f(i)}$;
2. $d = (d \oplus a) \ggg 32$;
3. $c = c \boxplus d$;
4. $b = (b \oplus c) \ggg 24$;
5. $a = a \boxplus b \boxplus m_{g(i)}$;
6. $d = (d \oplus a) \ggg 16$;
7. $c = c \boxplus d$;
8. $b = (b \oplus c) \ggg 63$.

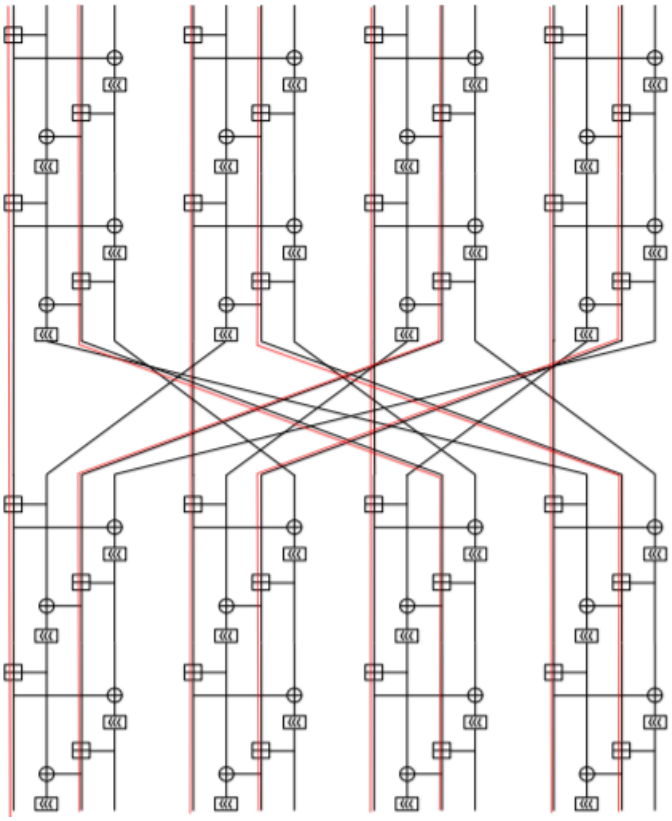


Рисунок 1.1 – Один раунд перестановки у геш-функції BLAKE2

SPECK

Загальна схема сім'ї блокових шифрів SPECK була розроблена АНБ (Агентством національної безпеки) у 2013 році [9] з метою протидії атакам за відкритим текстом та відкритим шифротекстом, представників якої записують як SPECK$2n/nm$, де $2n$ є розміром блоку у бітах для $n \in \{16,24,32,48,64\}$, а nm є розміром ключа у бітах для $m \in \{2,3,4\}$. На рисунку 1.2 наведена візуалізація одного раунду такого шифрування для $m = 4$. У математичному представленні раундова функція має вигляд

$$(x^{(i+1)},y^{(i+1)}) = F_{k_i}(x^{(i)},y^{(i)}) = (f_{k_i}(x^{(i)},y^{(i)}),f_{k_i}(x^{(i)},y^{(i)}) \oplus (y_{(i)} \lll \beta)),$$

де $x^{(i)}$ та $y^{(i)}$ є розбиттям стану (вхідного повідомлення) на дві частини довжини n, k_i є раундовим ключем також довжини n, а функція

$$f_{k_i}(x^{(i)},y^{(i)}) = ((x^{(i)} \ggg \alpha) \boxplus y^{(i)}) \oplus k_i.$$

Відповідно $x^{(i+1)}$ та $y^{(i+1)}$ є вихідними значеннями раундової функції, кожне з яких довжини n. Схема розгортання ключів на кожному раунді працює таким чином:

$$k_{i+1} = f_{ct}(l_i,k_i) \oplus (k_i \lll \beta),\ l_{i+m-1} = f_{ct}(l_i,k_i), \text{ де } l_i, k_0 \in \mathbb{F}_{2^n}.$$

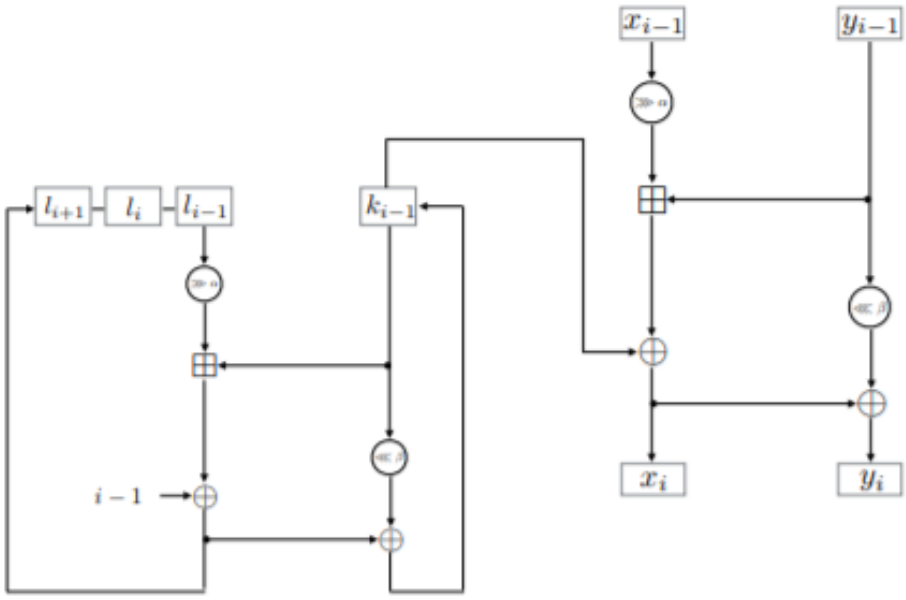


Рисунок 1.2 – Один раунд шифрування SPECK

ChaCha

У 2008 році Даніель Джуліус Бернштейн (англ. Daniel Julius Bernstein) запропонував схеми шифрування Salsa20 та ChaCha. Спочатку представив шифр Salsa20 [12], а потім — шифр ChaCha як модифікацію Salsa20 [13], метою якого було збільшення лавинних ефектів та покращення ефективності реалізації на деяких архітектурах.

Розглянемо функцію перестановки шифру ChaCha, до якої подається вектор x довжиною 512 біт, який представляється як матриця 4 на 4 байти, кожний елемент якої є двійковим вектором довжини n=32 біти, тобто

$$x = \begin{pmatrix} x_{0,0} & x_{0,1} & x_{0,2} & x_{0,3} \\ x_{1,0} & x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,0} & x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,0} & x_{3,1} & x_{3,2} & x_{3,3} \end{pmatrix} \in \mathbb{M}_{4x4}(\mathbb{F}_{2^n}).$$

Перестановка має 20 раундів функції *quarterround* Q, яка застосовується 10 разів до стовпців та 10 разів до діагоналей матриці x. У такій ситуації *quarterround* приймає чотири 32-бітові вектори x_0,x_1,x_2,x_3 та повертає чотири 32-бітові вектори y_0,y_1,y_2,y_3.

Якщо

$$(y_{0,i},y_{1,i},y_{2,i},y_{3,i}) = Q(x_{0,i},x_{1,i},x_{2,i},x_{3,i}), \text{ де } i = 0,1,2,3,$$

то кажуть, що *quarterround* застосована до стовпців.

Якщо

$$(y_{0,i},y_{1,i+1},y_{2,i+2},y_{3,i+3}) = Q(x_{0,i},x_{1,i+1},x_{2,i+2},x_{3,i+3}), \text{ де } i = 0,1,2,3,$$

то кажуть, що *quarterround* застосована до діагональних елементів матриці.

Математично робота функції Q записується аналогічно до функції G у BLAKE2, але з іншими константами циклічного зсуву.

Кипарис

У 2017 році у роботі [15] Марія Родінко та Роман Олійников запропонували симетричний блоковий шифр, який розроблений для малоресурсних пристроїв. Криптоаналіз такого шифру є ключовим у даній роботі, тому наведемо детальний його опис.

Специфікація шифру «Кипарис» передбачає дві можливі реалізації: для 32-бітової та 64-бітової архітектури, тому виділяють «Кипарис-256» та «Кипарис-512». Останнє число у назві відповідає довжині блоку відкритого тексту у бітах, що подається на вхід до функції шифрування, а також довжині ключа. Довжина слова складає 32 біти для «Кипарис-256» та 64 біти для «Кипарис-512», а відповідна кількість раундів 10 та 14. Ці та інші параметри шифру наведені у таблиці 1.1.

Таблиця 1.1 – Параметри шифру «Кипарис» [15]

	Кипарис-256	Кипарис-512
Розмір блоку n у бітах	256	512
Довжина ключа k у бітах	256	512
Довжина слова s у бітах	32	64
Кількість раундів t	10	14
Константи зсуву (r_1,r_2,r_3,r_4)	(16, 12, 8, 7)	(32, 24, 16, 15)

На рисунку 1.3 представлена схема шифрування «Кипарис», на вхід до якої подається блок відкритого тексту довжиною 8 на s біт $x = x_0 \parallel ... \parallel x_7$ і раундові ключі $RK^{(0)}$,...,$RK^{(t-1)}$, кожний з яких є ключем довжиною 4 на s біт, тобто

$$RK^{(i)} = RK_0^{(i)} \parallel RK_1^{(i)} \parallel RK_2^{(i)} \parallel RK_3^{(i)}, \text{ де } i = \overline{0,t-1}.$$

Розглядаються дві частини відкритого тексту x:

$$L_0 = x_0 \parallel x_1 \parallel x_2 \parallel x_3 \text{ та } R_0 = x_4 \parallel x_5 \parallel x_6 \parallel x_7.$$

Вихід з i-того раунду шифрування обчислюється так:

$$L_i = R_{i-1} \oplus F(L_{i-1}, RK^{(i-1)}),\ R_i = L_{i-1}.$$

Функція F є двократним застосуванням ARX-функції $h(x_0^{'} \parallel x_1^{'} \parallel x_2^{'} \parallel x_3^{'})$ до результату додавання L_{i-1} до ключа $RK^{(i-1)}$ за модулем 2. Функція $h(x_0^{'} \parallel x_1^{'} \parallel x_2^{'} \parallel x_3^{'})$ працює таким чином:

1. $x_0^{'} = x_0^{'} \boxplus x_1^{'}, x_3^{'} = x_3^{'} \oplus x_0^{'}, x_3^{'} = x_3^{'} \lll r_1$;
2. $x_2^{'} = x_2^{'} \boxplus x_3^{'}, x_1^{'} = x_1^{'} \oplus x_2^{'}, x_1^{'} = x_1^{'} \lll r_2$;
3. $x_0^{'} = x_0^{'} \boxplus x_1^{'}, x_3^{'} = x_3^{'} \oplus x_0^{'}, x_3^{'} = x_3^{'} \lll r_3$;
4. $x_2^{'} = x_2^{'} \boxplus x_3^{'}, x_1^{'} = x_1^{'} \oplus x_2^{'}, x_1^{'} = x_1^{'} \lll r_4$.

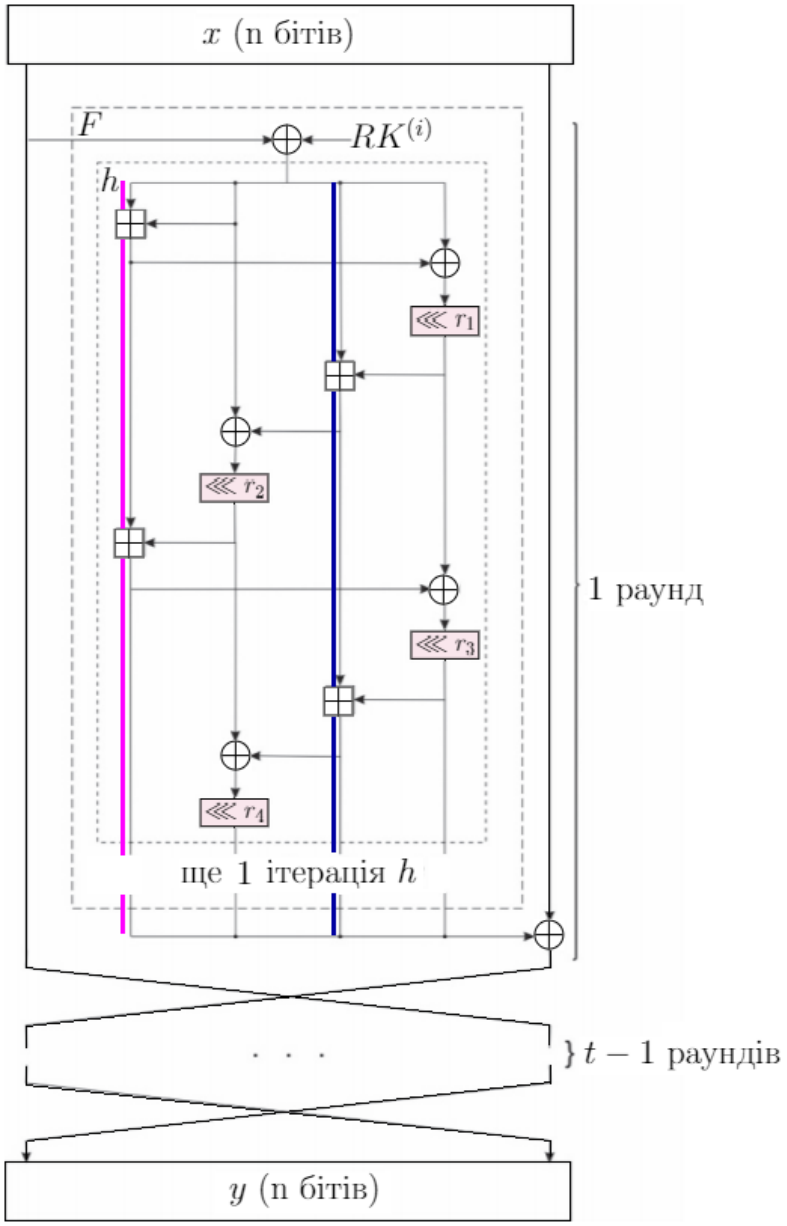


Рисунок 1.3 – Функція шифрування «Кипарис»

Раундові ключі формуються у три етапи, використовуючи ключ $K = K_0 \parallel ... \parallel K_7$, де довжина K_j $(j = \overline{0,7})$ складає s біт і константу $tmv = 0x000F000F \parallel 0x000F000F \parallel 0x000F000F \parallel 0x000F000F$.

1 етап. На цьому етапі формуємо допоміжний ключ K_σ довжиною 4 на s біт, використовуючи ключ $K = K_0 \parallel ... \parallel K_7$:

1. $K_l = K_0 \parallel K_1 \parallel K_2 \parallel K_3$, $K_r = K_4 \parallel K_5 \parallel K_6 \parallel K_7$;
2. $st = 0x1 \oplus K_l$, $st = h(h(st))$;
3. $st = st \boxplus K_r$, $st = h(h(st))$;
4. $st = st \oplus K_l$, $K_\sigma = st$.

2 етап. Використовуючи ключ шифрування K, допоміжний ключ K_σ та константу tmv виконуємо такі операції:

1. $st = K_0 \parallel K_1 \parallel K_2 \parallel K_3$, $K_t = K_\sigma \boxplus tvm$;
2. $st = st \boxplus K_t$, $st = h(h(st))$;
3. $st = st \oplus K_t$, $st = h(h(st))$;
4. $st = st \boxplus K_t$, $RK^{(2i)} = st$;
5. $tmv = ShiftLeft(tmv,0x1)$, де $ShiftLeft$ — зсув кожного s-бітового слова вектора tmv на 1 біт вліво;
6. $st = K_4 \parallel K_5 \parallel K_6 \parallel K_7$, $K_t = K_\sigma \boxplus tvm$;
7. $st = st \boxplus K_t$, $st = h(h(st))$;
8. $st = st \oplus K_t$, $st = h(h(st))$;
9. $st = st \boxplus K_t$, $RK^{(2i+1)} = st$.

3 етап. Після отримання ключів на парному і непарному раундах на другому етапі, оновлюємо константу tmv та ключ шифрування K:

1. $ShiftLeft(tmv,0x1)$;
2. $K = ROTLKey(K,0x1)$, де $ROTLKey$ — циклічний зсув s-бітових слів ключа шифрування K на 1 біт вліво;
3. Повертаємось до другого етапу.

У 2020 році Марія Родінко та Роман Олійніков дослідили ефективність реалізації шифру «Кипарис» на різних платформах [16]. Для точності результатів дослідники робили восьмикратне зашифрування блоку пам'яті у режимі електронної кодової книги на мові C++. Блок

пам'яті складався з N блоків даних фіксованого розміру, що представлені у вигляді масивів 64-бітових невід'ємних цілих чисел. N та довжина масиву залежали від розміру вхідного блоку у шифр. Не є за потрібним перераховувати процесори, на яких відбувались заміри швидкодії, адже зрозуміло, що чим потужніший процесор, тим більшу кількість операцій він може виконати. Зазначимо тільки, що платформами, з якими працювали автори роботи [16], були Windows (32- та 64-бітові архітектури), Linux (64-бітова), Android (32- та 64-бітові). Відбувалось порівняння швидкодії «Кипарис-256» та «Кипарис-512» з AES-256, ГОСТ-28147-89:2009, SPECK64/128, SPECK128/128 та SPARX128/128. Для цього було обрано найефективніші реалізації зазначених шифрів. За результатами дослідження отримано, що процес зашифрування у «Кипарис» відбувається швидше у порівнянні з іншими схемами, а на деяких платформах приблизно на одному рівні з SPECK64/128 і SPECK128/128, що дало змогу зробити висновок про можливість використання шифру «Кипарис» у постквантовий період. Однак квантовим криптоаналізом даного шифру ніхто не займався, тому у третьому розділі даної роботи вперше дається цьому початок.

Зауважимо, що дуже часто під ARX-схемами мають на увазі не тільки схеми, що використовують операції додавання за модулем, виключної диз'юнкції та циклічного зсуву, а й схеми, де разом із зазначеними операціями (або замість деяких з них) використовують операції побітового додавання (AND), зсуву (shift), побітового або (OR). Прикладами таких схем є блокові шифри Simon [9], Simeck [17], схема аутентифікованого шифрування NORX [18], а також сім'я геш-функцій MD4 [19].

Алгоритми легкої криптографії можуть розроблятись як абсолютно нові з метою зниження витрат на апаратну реалізацію, так і бути певними модифікаціями вже наявних стандартизованих алгоритмів для персональних комп'ютерів, але з певними оптимізаціями. Відповідно виникло питання: чи будь-яку функцію можна подати як комбінацію

зазначених вище операцій, відповідь на яке позитивна. У 2010 році Дмитро Ховратовіч (англ. Dmitry Khovratovich) та Івіца Ніколіч (англ. Ivica Nikoli´c) показали, що функціонально повною системою для реалізації будь-якої функції є $\{\boxplus, \oplus, \lll, 1\}$. Також вони показали еквівалентність цієї множини базису $\{\boxplus, \lll, 1\}$, що стало поштовхом для аналізу стійкості потенційних схем шифрування лише з операціями додавання за модулем та циклічного зсуву (з можливими константами). Дмитро Ховратовіч і Івіца Ніколіч отримали, що з базисом $\{\boxplus, \lll, 1\}$ будь-яка схема шифрування (далі AR-схема) може бути апроксимована до лінійної функції з ймовірністю 2^{-q}, де q — кількість додавань у схемі, що робить схему шифрування вразливою до лінійного криптоаналізу, якщо кількість додавань (відповідно і раундів) мала. Більш того, якщо маємо RX-схему (тобто схему шифрування лише з операціями циклічного зсуву та виключної диз'юнкції), то її легко зламати, розв'язуючи систему лінійних рівнянь за модулем 2 (що можна зробити ефективно) [20]; AX-схему також можна зламати за поліноміальний час [21].

1.2 Методи криптоаналізу ARX-шифрів

У той час коли криптографи займаються розробкою криптографічних методів захисту інформації, які б забезпечували конфіденційність та цілісність даних, криптоаналітики досліджують ці методи та отримують обґрунтовані оцінки їх стійкості. Під стійкістю розуміють здатність криптопримітива не бути зламаним під дією атак криптоаналітика. У 1883 році Огюст Керкгоффс (англ. Auguste Kerckhoffs) у своїй монографії «Воєнна криптографія» сформулював правило, яке каже, що стійкість криптопримітива не повинна визначатись незнанням криптоаналітика способу шифрування, а повинна визначатись лише його незнанням ключа шифрування та якістю шифрування. Дане правило і сьогодні є основним припущенням у криптоаналізі. У залежності від того, яка інформація відома криптоаналітику, виділяють

такі види криптоаналізу:

1) *Криптоаналіз на основі шифротексту*, коли криптоаналітик знає шифротекст або декілька шифротекстів, що отримані з допомогою алгоритму шифрування, на який криптоаналітик планує зробити атаку. Його задача у такому випадку полягає у знаходженні відкритого тексту, або у знаходженні секретного ключа (чи ключів) шифрування, або у побудові алгоритму розшифрування якомога більшої кількості шифротекстів.

2) *Криптоаналіз на основі відкритого тексту*, коли криптоаналітику відомі пари (ВТ, ШТ), де ВТ — відкритий текст, а ШТ — шифротекст. Задача криптоаналітика у такому випадку полягає у знаходженні секретного ключа (чи ключів) шифрування або у побудові алгоритму розшифрування якомога більшої кількості шифротекстів.

3) *Криптоаналіз на основі вибраного відкритого тексту*, коли криптоаналітик вибирає довільний відкритий текст, знаходить відповідний йому шифротекст і прагне знайти секретний ключ (чи ключі) шифрування або побудувати алгоритм розшифрування якомога більшої кількості шифротекстів.

4) *Адаптивний криптоаналіз на основі вибраного відкритого тексту* — криптоаналіз на основі вибраного відкритого тексту, коли криптоаналітик вибирає наступний відкритий текст спираючись на отримані результати шифрування.

5) *Криптоаналіз на основі вибраного шифротексту*, коли криптоаналітик має доступ до певного алгоритму розшифрування і прагне знайти секретний ключ (чи ключі) шифрування.

6) *Криптоаналіз з використанням вибраного ключа*, коли криптоаналітик знає деяку інформацію про зв'язок між ключами криптопримітива, але не знає самі ключі.

7) *Криптоаналіз з використанням додаткової інформації*, яка отримана криптоаналітиком з побічного каналу або шляхом підкупу (чи шантажу).

8) А також *криптоаналіз з використанням різних комбінацій зазначених вище видів* [22].

Існує велика кількість різноманітних методів та підходів до криптоаналізу, починаючи з повного перебору, парадоксу про день народження та зустрічі посередині, закінчуючи комбінаціями методів диференціального, лінійного, ротаційного криптоаналізів та їх модифікаціями. Побоювання щодо створення достатньо потужного квантового комп'ютера стало поштовхом для виникнення ще одного напряму — квантового криптоаналізу. Найпоширенішими методами, що є застосовними до ARX-шифрів, сьогодні є диференціальний, лінійний та ротаційний криптоаналізи.

Диференціальний криптоаналіз

У своїй статті 1990 року [23] Елі Біхам (англ. Eli Biham) і Аді Шамір (англ. Adi Shamir) представили світу такий метод криптоаналізу як диференціальний криптоаналіз, який є статистичною атакою за вибраним відкритим текстом на блокові шифри, застосували його до DES, а у 1991 році до шифру FEAL [24]. Так з'явилось поняття *диференціала* (або *різниці*) пари відкритих (x,x') та шифрованих (y,y') текстів, тобто

$$\text{пари } (\alpha,\beta), \text{ де } \alpha = x \circledast (x')^{-1} \text{ та } \beta = y \circledast (y')^{-1},$$

Значення $(x')^{-1}$ та $(y')^{-1}$ — обернені до x' та y' відповідно за операцією $\circledast$. За замовчуванням у якості операції $\circledast$ виступає $\oplus$. Відповідно розглядають

$$\text{пари } (\alpha,\beta), \text{ де } \alpha = x \oplus x' \text{ та } \beta = y \oplus y'.$$

Для того, щоб провести диференціальний криптоаналіз, досліджують послідовність різниць між двома текстами на кожному раунді шифрування і рахують відношення кількості повторень найчастішої різниці у шифрі до загальної кількості різниць. Таку послідовність називають *диференціальною характеристикою*, а відношення — *ймовірністю диференціальної характеристики*. Також вводять поняття *вкладеної диференціальної характеристики*, тобто диференціальної характеристики, що є частиною іншої диференціальної характеристики,

яка містить першу, а також ще по одній різниці на початку та на кінці.

Метою диференціального криптоаналізу є знаходження диференціала (α,β), який зустрічається набагато частіше, аніж інші диференціали, тобто пошук таких α та β, що ймовірність зустрічі пари відкритих (x,x') та шифрованих (y,y') текстів з диференціалом (α,β) є найвищою. Якщо така пара (α,β) знаходиться і ймовірність відрізняється від випадкової перестановки, то в багатьох випадках можна повністю відновити секретний ключ або зламати шифр. Варто зазначити, що якщо говорять про диференціальний криптоаналіз, то зазвичай розглядають ітеративні шифри, тобто

$$y(i) = f(y(i-1),k_i),$$

де $y(0) = x$ — заданий відкритий текст, $i > 1$, k_i — ключ на i-му кроці, а f — раундова функція шифрування. Більш того, для проведення атаки розглядають диференціальну характеристику функції шифрування без її останнього раунду.

Так, наприклад, якщо є ітеративний шифр з r раундами шифрування і диференціалом (α,β), то його диференціальною характеристикою є

$$\alpha = \Delta y(0), \Delta y(1), \dots, \Delta y(r) = \beta,$$

де $\Delta y(i)$ — різниця на i-тому раунді шифрування. Якби такий шифр утворював ланцюг Маркова, тоді можна було б розглядати раунди шифрування як незалежні та рахувати ймовірність диференціальної характеристики як добуток ймовірностей на кожному раунді. Однак не все так просто, тому у 1991 році Сюецзя Лай (англ. Xuejia Lai) та Джеймс Мессі (англ. James Lee Massey) у своїй статті [25] дослідили умови, при яких диференціальні характеристики утворюють ланцюг Маркова. Так з'явилось поняття *шифру Маркова* і відповідна теорема на випадок коли шифр є марківським.

Лінійний криптоаналіз

У 1992 році Міцуру Мацуї (англ. Mitsuru Matsui) та Ацухіро Ямагіші (англ. Atsuhiro Yamagishi) написали роботу [26], у якій

запропонували новий метод криптоаналізу, вдало застосувавши його до різних версій шифру FEAL. Цікаво, що такий застосунок був прикладом реалізації лінійного криптоаналізу, але його формалізацію Міцуру Мацуї опублікував тільки через рік [27], де надав першу атаку на шифр DES (англ. Data Encryption Standard), яка довгий час була найкращою.

Основною метою лінійного криптоаналізу є заміна нелінійної функції у схемі шифрування лінійною, що очевидним чином спрощує подальший аналіз, якщо значення лінійного наближення збігаються зі значеннями нелінійної функції найбільшу кількість разів. Для цього, окрім самого шифру, треба мати ще достатню кількість пар (ВТ, ШТ). Таким чином маємо справу з криптоаналізом на основі відкритого тексту. Лінійне наближення розглядають як

$$\alpha \cdot X \oplus \beta \cdot Y = \gamma \cdot K$$

де X, Y, K — ВТ, ШТ та значення ключа, α, β, γ — константи, а $\cdot$ — операція скалярного добутку.

Міцуру Мацуї у своїй роботі [27] сформулював два алгоритми. Перший алгоритм є алгоритмом, що встановлює один біт ключа K, а другий — декілька. Також навів декілька лем, серед яких лема про оцінку надійності роботи лінійного криптоаналізу через функцію нормального розподілу та знаменита лема про набігання знаків (англ. piling-up lemma).

Ротаційний криптоаналіз

Класичними методами криптоаналізу ARX-шифрів є лінійний та диференціальний криптоаналізи. Однак у 2010 році Дмитро Ховратович і Івіца Ніколіч запропонували новий — ротаційний криптоаналіз (R-криптоаналіз) [20], який є ймовірнісною атакою за вибраним відкритим текстом. Розглядались пари текстів

$$x \text{ та } x' = x \lll \gamma,$$

де γ є деякою константою, яку дослідники назвали *параметром ротації*, а саму пару (x,x') — *парою ротації на* γ.

Ідея R-криптоаналізу ґрунтується на тому, що якщо для довільно

обраної пари ротації відкритих тестів (x,x') відповідна вихідна пара (y,y') ARX-схеми також є парою ротації (далі R-парою), то з ймовірністю вище, ніж для випадкової перестановки, можна сказати який був відкритий текст за шифротекстом. Для цього дослідники згадують роботу Магнуса Даума (англ. Magnus Daum) [19], в якій запропонована формула обрахунку ймовірності ротації для одного додавання, і розглядають ймовірності виходу кожного раунду шифрування ARX-схеми так само як і у диференціальному криптоаналізі. Однак у диференціальному криптоаналізі Сюецзя Лай та Джеймс Мессі показали, що обчислення ймовірності виходу криптографічного примітива, як добутку ймовірностей на кожному раунді шифрування, є можливим лише за припущенням про незалежність та рівноймовірність ключів, адже тоді різниці після кожного раунду шифрування можна розглядати як ланцюг Маркова [25]. Аналогічне припущення зробили Дмитро Ховратович і Івіца Ніколіч для R-криптоаналізу у своїй статті 2010 року [20], а у 2015 році опублікували статтю [28], де показали, що таке припущення не є правильним, сформулювали теорему про послідовні додавання за модулем, застосували її до геш-функцій BLAKE2 та Skein, а також перевірили результати експериментально, використовуючи комп'ютерну симуляцію. Атака, що зроблена дослідниками на Skein, є найкращою сьогодні, але окрім R-криптоаналізу, вона використовує ще й диференціальний криптоаналіз (маємо атаку відскоку) [28, 29].

Дмитро Ховратович і Івіца Ніколіч розглядали роботи, що є застосовними саме до ARX-схем, а не до ARX-схем з введеними константами. Розробники блокового шифру SEA помітили, що SEA є стійким до криптоаналізу циклічного зсуву саме через введення псевдовипадкових констант у схему розгортання ключів [30], що говорить про необхідність глибшого вивчення R-криптоаналізу на такий випадок.

Статті [20, 28] стали поштовхом до ARX-криптоаналізу, яким зацікавились Томер Ашур (англ. Tomer Ashur) та Янвен Лю (англ. Yunwen Liu). Вони запропонували новий підхід — криптоаналіз

циклічного зсуву та виключної диз'юнкції (далі RX-криптоаналіз), у якому константи вводяться у стан (вхідне повідомлення), і у 2016 році опублікували статтю [31]. Пару відкритих текстів (x_1,x_2), для яких

$$x_1 = x \oplus a_1 \text{ та } x_2 = (x \lll \gamma) \oplus a_2,$$

де γ, a_1, a_2 є фіксованими значеннями, запропонували називати *RX-парою для деякого відкритого тексту x та фіксованих значень* γ, a_1, a_2, а пару ((a_1,a_2),γ) — *RX-різницею*. Скориставшись теоремою Хельгер Ліпмаа (англ. Helger Lipmaa) та Шіхо Мopiai (англ. Shiho Moriai) для диференціальної ймовірності додавань [32], дослідники сформулювали та довели теорему про ротацію з константами, але тільки для одного додавання. На прикладі блокового шифру SPECK у роботі [31] перевірили справедливість теореми експериментально для версії з $n = 16, m = 4, \alpha = 7, \beta = 2$, а у роботі [33] разом із Гленн де Віттом (англ. Glenn De Witte) та Адріаном Ранеа (англ. Adrián Ranea) для старших версій.

Томер Ашур, Янвен Лю та Адріан Ранеа продовжили досліджувати RX-криптоаналіз і у 2020 році опублікували статтю [34], у якій описали метод пошуку оптимальної RX-характеристики, використовуючи SAT-вирішувач. У своїй роботі вони подали еквівалентне представлення теореми про ротацію з константами для одного додавання, ввели означення *RX-характеристики* та *ваги RX-характеристики*.

Цікавою вийшла стаття [35] 2020 року Стефано Барберо (англ. Stefano Barbero), Емануеля Белліні (англ. Emanuele Bellini) та Рузіді Макаріма (англ. Rusydi Makarim), головною метою яких було застосувати ротаційний криптоаналіз до функції перестановки шифру ChaCha. Дослідники сформулювали теорему, де, на відміну від теореми про послідовні додавання за модулем у статті [28], не вимагали збереження послідовних різниць на кожному раунді шифрування, а також у доведенні й мови не було про залежність чи незалежність виходів із раундових функцій. Отримано, що обрахунки за такою теоремою і теоремою про послідовні додавання за модулем повністю збігаються у тих випадках,

коли параметр ротації γ лежить у проміжку $(1, n-1)$, де n — довжина вхідного (і відповідно вихідного) слова, а при $\gamma = 1$ або $\gamma = n-1$ ймовірність є вищою ніж при використанні теореми про послідовні додавання за модулем. Також у роботі [35] встановлені оцінки верхньої та нижньої границь ймовірностей ротації для функції перестановки ChaCha, спираючись на незалежність виходів підфункції *quarterround*.

Згодом у статті [36] Лілія Кралева (англ. Liliya Kraleva), Томер Ашур та Вінсент Раймен (англ. Vincent Rijmen) застосували ротаційний криптоаналіз до Chaskey. Також у червні 2020 року Бонвук Ку (англ. Bonwook Koo), Янгхун Юнг (англ. Younghoon Jung) та Ву-Хван Кім (англ. Woo-Hwan Kim) у роботі [37] поширили RX-криптоаналіз на випадок, коли замість додавання за модулем у схемі шифрування розглядається побітове додавання (маємо так звану AND-RX-схему) на прикладі шифру Simon, а у серпні Джиню Лу (англ. Jinyu Lu), Янвен Лю, Томер Ашур, Бінг Сан (англ. Bing Sun) та Чао Лі (англ. Chao Li) опублікували роботу [38], де представили аналогічні результати до [37], а також застосували їх до Simeck.

Детальніший аналіз елементів робіт з ротаційного криптоаналізу ARX-шифрів буде наведено у другому розділі.

Квантовий криптоаналіз

Перед тим, як перейти до детального опису елементів квантового криптоаналізу, треба обґрунтувати необхідність створення квантового комп'ютера, адже саме так стане зрозуміло чому такі компанії як D-Wave Systems, Google, NASA, Intel, IBM та інші вкладають величезні кошти на його створення. Для цього можна згадати першу квантову революцію, під час якої світ дізнався про такі речі як провідники та транзистори, нанотехнології та ядерну енергетику. Цей період дав можливість зрозуміти, чому зірки сяють, зрозуміти структуру ДНК, а також вступити в епоху молекулярної біології. Зараз відбувається друга квантова революція, головним героєм якої є квантовий комп'ютер [39, 40, 41]. Сьогодні все частіше виникають задачі, які вимагають певних

оптимізацій чи повного перебору величезної кількості даних. Так дуже популярною темою є використання нейронних мереж і штучного інтелекту в аналізі даних із супутників та даних медичного характеру. Наприклад, на Kaggle є змагання, у якому пропонується побудувати розпізнавач, який за картинками легень може характеризувати тип пневмонії [42]. Зрозуміло, що на точність роботи такого розпізнавача впливає не тільки знання з машинного навчання, а й кількість інформації, на якій розпізнавач, тобто нейрона мережа, вчиться. Так можна припустити, що квантовий комп'ютер може відіграти велику роль у медицині. Також рахунки за електроенергію можуть зменшитись, адже тоді вдасться знайти високотемпературні надпровідники, які проводять енергію без опору.

Першим, хто висловив ідею щодо створення квантового комп'ютера був Річард Фейнман (англ. Richard Feynman, 1982 рік) [43]. Через три роки ввели поняття *квантової машини Тюрінга* [44], а у 1998 році вперше у світі була представлена робота двокубітового квантового комп'ютера. Нещодавно компанія D-Wave Systems представила вже комп'ютер з 5000 кубітами [45]. Досить довго ніхто не бачив у цьому загрози для криптографії, більшість ставилась до створення такого комп'ютера скептично. Однак тепер вже всім стало зрозуміло, що це все може стати реальністю.

Можна вважати, що перший крок у квантовому криптоаналізі зробив у 1994 році Пітер Шор (англ. Peter Shor) у роботі [46], адже він показав як можна розкласти будь-яке число N на прості множники за час $O(\log^3 N)$, використовуючи при цьому $O(\log N)$ кубітів. Стало очевидно, що при створенні потужного квантового комп'ютера асиметричні криптопримітиви втратять свою надійність, адже їх підґрунтям є задача факторизації або задача дискретного логарифмування, які у класичній моделі обчислень мають принаймні субекспоненційну складність.

Більш того, є певні сумніви щодо стійкості симетричних криптопримітивів. У 1993 році Ітан Бернштейн (англ. Ethan Bernstein) та

Умеш Вазірані (англ. Umesh Vazirani) сформували задачу пошуку двійкового вектора скалярного добутку та алгоритм розв'язку з одним запитом до квантового оракула [47]. Через рік Даніель Саймон (англ. Daniel Simon) у роботі [48] представив задачу пошуку періоду функції та показав, що вона розв'язується у квантовій моделі за поліноміальний час, а у 1996 році Лов Гровер (англ. Lov Grover) зменшив час розв'язку задачі пошуку унікального елементу в невпорядкованій базі даних до субекспоненційної у квантовій моделі [49]. Ці задачі відносно довгий час не цікавили нікого з точки зору симетричної криптографії, поки у 2010 році Кувакадо Хіденорі (англ. Kuwakado Hidenori) та Мораї Масакату (англ. Morii Masakatu) у статті [50] не запропонували шлях зведення атаки розрізнення випадкового тексту від шифротексту для класичної трираундової схеми Фейстеля до задачі Саймона. Згодом запропонували атаки на узагальнені схеми Фейстеля типу I та II [51], атаку на схему Івена-Мансура (англ. Even–Mansour), слайд-атаки на коди аутентифікації [52], атаки на FX-конструкції та класичну п'ятираундову схему Фейстеля з об'єднанням алгоритмів Гровера та Саймона [53, 54], а також атаки на схеми Фейстеля, де відбувається чергування ключів [55]. Окрім цього, з'явились квантовий диференціальний та лінійний криптоаналізи симетричних схем шифрування [56, 57, 58, 59].

Елементи робіт з квантового криптоаналізу буде наведено у другому розділі.

Інші методи криптоаналізу, їх модифікації та комбінації

Очевидно, що класичні, «прямі» методи криптоаналізу не завжди дають бажаний результат, а також не завжди є достатня кількість ресурсів для їх проведення. Так, наприклад, для реалізації диференціальної атаки розглядають максимальну ймовірність диференціальної характеристики, тобто шукають максимум за усіма вкладеними характеристиками від середньої за ключами диференціальної характеристики. Зрозуміло, що якщо мають справу з шифрами, де розмір блоку і кількість раундів є великим числом, то пошук такої

характеристики є практично неможливою сьогодні задачею, адже для цього потрібно перебрати експоненційну кількість характеристик. До речі саме через це вводять поняття практичної та теоретичної стійкості, перша з яких є максимальною ймовірністю диференціальної характеристики, а друга — максимальною ймовірністю диференціала. Таким чином розробляються методи, які допомагають аналізувати шифри й у таких випадках. Найпопулярнішим методом оцінки ймовірності диференціальної характеристики ARX-шифрів, що зараз застосовується, є модифікований алгоритм Мацуї, де замість усіх диференціалів, перебирають обрані, тобто ті, що не менше певної константи. Такий метод диференціального криптоаналізу застосувала Марія Родінко у своїй роботі [60] до шифру «Кипарис».

Існують й інші методи. Одним із найпоширеніших є метод бумеранга (атака бумеранга), який є варіантом диференціальної атаки. Однак у такому методі усю схему шифрування розглядають не як одну схему з певною кількістю раундів, а як дві схеми шифрування з деякою меншою кількістю раундів. Тобто схему шифрування «поділяють» на дві частини, які необов'язково є рівними, і застосовують диференціальні атаки до кожної з цих частин. Ідея такої атаки вперше з'явилась у Девіда Вагнера (англ. David Wagner) у роботі 1999 року [61], а поштовхом до неї стала думка про те, що для того, щоб унеможливити атаки диференціального типу на шифри, треба між деякими раундами схеми шифрування реалізовувати декореляційні модулі, тобто функції, які знищують ймовірності диференціалів. Девід Вагнер показав, що така думка є хибною і застосував атаку бумеранга до блокового шифру COCONUT98, а також зазначив, що шифр FEAL є вразливим до даної атаки. Такий підхід потребує знання не тільки відкритого тексту, а й знання функції розшифрування, тому не завжди може застосовуватись до шифрів. Підсилена атака бумеранга, яка є модифікацією класичної атаки бумеранга, але не потребує знання функції розшифрування, була представлена у 2000 році у роботі [62] Джона Келсі (англ. John Kelsey),

Тадайосі Коно (англ. Tadayoshi Kohno) та Брюса Шнайєра (англ. Bruce Schneier). У 2001 році Елі Біхам, Орр Данкелмен (англ. Orr Dunkelman) та Нейтан Келлар (англ. Nathan Keller) запропонували ще одну модифікацію атаки бумеранга під назвою атака прямокутника (англ. rectangle attack) [63].

Дуже ефективними вважаються комбінації диференціального та лінійного криптоаналізів. Тобто атаки, в яких перша частина шифру розглядається з точки зору диференціального криптоаналізу, а інша — лінійного. Вперше ця ідея з'явилась у Сьюзен Лангфорд (англ. Susan Langford) та Мартіна Геллмана (англ. Martin Hellman) у 1994 році [64] і з того часу зазнала різних модифікацій. Зокрема у 2020 році вченими запропоновано підхід саме до ARX-криптосистем, у якому покращена лінійна частина атаки, завдяки чому, для знаходження ключа достатньо просто застосувати швидке перетворення Уолша-Адамара [65]. Також у березні 2021 року Янвен Лю та її колеги опублікували роботи [66, 67], в яких запропонували комбінацію RX-криптоаналізу та лінійного, але, як це і зазначили дослідники, не все у цьому напряму вдалось зробити. Наприклад, одного припущення про незалежність частини функції шифрування, яка розглядається з точки зору RX-криптоаналізу, і частини, яка розглядається з точки зору лінійного криптоаналізу, недостатньо для того, щоб отримати формулу обрахунку *зміщення* (англ. bias), тобто параметра, який показує на скільки ймовірність проведення атаки відрізняється від 0.5.

Слайд-атаки у 1999 році запропонували Алекс Бірюков (англ. Alex Biryukov) та Девід Вагнер у роботі [68]. Метод цікавий тим, що кількість раундів у шифрі ніяк не впливає на успішність проведення атаки, а визначається стійкістю раундової функції. Для цього шукають *слайдову пару*, тобто пару, у якій другий елемент отриман з першого шляхом застосування раундової функції; накопичують достатню кількість статистики (слайдових пар) та застосовують інші методи криптоаналізу.

Ще один метод криптоаналізу — криптоаналіз на пов'язаних ключах,

який запропонував у 1993 році Елі Біхам у статті [69], де досліджувалась стійкість блокових шифрів, враховуючи вплив схем розгортання ключів. Успішність проведення атаки визначається у даному методі лише стійкістю схеми розгортання ключів та раундової функції, а сам алгоритм дій нагадує слайдову атаку. Зазвичай цей метод неможливо реалізувати на практиці, адже пошук слайдової пари у такому методі є еквівалентним за складністю до задачі про день народження (це видно з системи рівнянь у статті [69]), але існують шифри, для яких дана атака є успішною, наприклад, LOKI89, LOKI91, Lucifer [69].

Також можна згадати про алгебраїчний криптоаналіз, основною метою якого є побудова системи двійкових рівнянь, що пов'язує вхід у схему шифрування з її виходом та ключем, а також вирішення цієї системи, що зазвичай є NP-повною задачею. Для цього використовують методи лінеаризації двійкових функцій та узагальнений метод Гаусса. Однак сьогодні не існує прикладів застосування алгебраїчного криптоаналізу до ARX-шифрів, що пов'язано зі складним математичним підґрунтям.

Висновки до розділу 1

Нас оточує багато пристроїв з обмеженими обчислюваними можливостями та пам'яттю (мобільні телефони, смарткартки, RFID-мітки й т.д.), а з розвитком Інтернету речей їх кількість з кожним роком тільки збільшуватиметься. Криптосистеми, які використовують на сучасних персональних комп'ютерах для захисту інформації, часто не можна ефективно реалізувати на малоресурсних системах, тому виникає необхідність розробки швидких алгоритмів шифрування, враховуючи баланс між безпекою та ресурсами. ARX-шифри справляються з такою задачею краще всіх, але питання безпеки все одно залишається відкритим. У даному розділі розглянуті наявні ARX-схеми, описані методи криптоаналізу, у тому числі й відносно нові підходи до

ARX-аналізу (R- та RX-криптоаналізи), а також квантовий криптоаналіз, елементи яких детальніше будуть представлені у другому розділі даної роботи, а у третьому — застосовані до малоресурсного симетричного блокового шифру «Кипарис».

2 ТЕОРЕТИЧНІ ОСНОВИ КВАНТОВОГО І РОТАЦІЙНОГО КРИПТОАНАЛІЗІВ СИМЕТРИЧНИХ СХЕМ ШИФРУВАННЯ

У цьому розділі наведено основи квантового криптоаналізу симетричних схем шифрування: задачі Саймона та Гровера, застосувавши комбінацію алгоритмів розв'язків яких до класичної п'ятираундової схеми Фейстеля, побудована субекспоненційна оцінка складності атаки на неї, а також проведено дослідження стійкості узагальненої схеми Фейстеля типу I у квантовій моделі обчислень. Окрім цього, зроблено аналіз наявних сьогодні результатів з ротаційного криптоаналізу ARX-шифрів та розглянуті деякі приклади його застосування.

2.1 Квантовий криптоаналіз фейстель-подібних схем шифрування

Задача Саймона

Нехай для функції $f : \{0,1\}^n \to \{0,1\}^n$ існує єдине число a, що $f(x) = f(y)$ тоді та тільки тоді, коли $y = x \oplus a$. Задача Саймона полягає у тому, щоб знайти таке a.

Очевидно, що у класичному випадку розв'язок цієї задачі вимагає експоненційну кількість кроків та пам'яті. Однак у квантовій моделі обчислень Даніель Саймон показав, що складність пошуку a є поліноміальною [48]. Для того, щоб розглянути як йому вдалось це зробити, введемо деякі означення та твердження.

Означення 2.1. *n-кубітовим квантовим регістром* називають квантову систему, хвильова функція якої має вигляд

$$|\Psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle,$$

де амплітуди α_i задовільняють умові нормування

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1,$$

$|\cdot\rangle$ — кет-вектор як у нотації Дірака.

Важливо підкреслити, що коли говорять про квантові обчислення, то мають на увазі застосування унітарних операторів до векторів станів із метою їх зміни. Найвідомішим таким оператором є *оператор Адамара.*

Означення 2.2. *Оператором Адамара* називають унітарний оператор, який задається таким чином:

$$H_n = H_{n-1} \otimes H, \text{ де } H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

$\otimes$ — кронекеровий добуток матриць, n — розмірність квантової системи, до якої хочемо застосувати оператор Адамара.

Однак користуватись таким визначенням у загальному випадку незручно, тому застосовують альтернативне його представлення, що наведено нижче. Також у літературі дуже часто оператор Адамара називають «*перетворенням Адамара*» (англ. Hadamard transformation).

Твердження 2.1. *Оператор Адамара, що застосован до вектора $|x\rangle$ деякої системи з n кубітами, можна представити у вигляді:*

$$H_n |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} (-1)^{\langle x|y\rangle} |y\rangle,$$

де $\langle x|y\rangle$ є скалярним добутком векторів $\langle x|$ та $|y\rangle$.

Варто наголосити, що під квантовим оракулом розуміють формальну модель, що виконує роботу деякої функції f, внутрішня будова якої нам невідома. Тоді, спираючись на те, що у квантовій моделі обчислень все описується унітарними операторами, стає зрозуміло, що квантовий оракул повинен реалізовувати такий оператор U_f, який є оборотним. Зазвичай використовують *модель стандартного оракула*, який виглядає таким чином:

$$U_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|y \oplus f(x)\right\rangle.$$

Можна побачити, що U_f не впливає на значення $\left|x\right\rangle$, але використовує його для визначення значення функції $f(x)$, що обґрунтовує оборотність даного оператора. Також, підставляючи різні значення у регістри $\left|x\right\rangle$ та $\left|y\right\rangle$, легко впевнитись у реалізації усіх можливих значень функції $f(x)$.

Теорема 2.1. *Задача Саймона розв'язується у квантовій моделі обчислень за поліноміальну кількість кроків [48, 50, 51, 54].*

Доведення.

1) Задамо два квантових регістри, на вхід яким подамо n-кубітові нулі, тобто $\left|0\right\rangle^n \left|0\right\rangle^n$. Застосуємо перетворення Адамара до першого з них:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \left|x\right\rangle \left|0\right\rangle^n.$$

Таким чином отримали усі можливі вхідні дані.

2) Для того, щоб отримати значення функції $f(x)$, зробимо запит до квантового оракула, тобто застосуємо унітарний оператор U_f до двох регістрів:

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \left|x\right\rangle \left|f(x)\right\rangle.$$

3) Так як задача Саймона полягає у пошуку періоду функції a, то з умови маємо:

$$\frac{1}{\sqrt{2^n}} \sum_{\substack{x=0,\\ x\neq x\oplus a}}^{2^n-1} \left(\left|x\right\rangle + \left|x \oplus a\right\rangle\right) \left|f(x)\right\rangle.$$

4) Виміримо стан другого регістру й отримаємо значення функції f та суму його прообразів:

$$\frac{1}{\sqrt{2}}\left(\left|x\right\rangle + \left|x \oplus a\right\rangle\right) \left|f(x)\right\rangle.$$

5) Знову застосуємо перетворення Адамара до першого регістру:

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{\langle x|y\rangle} \left|y\right\rangle + \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{\langle x\oplus a|y\rangle} \left|y\right\rangle.$$

6) Якщо $\langle a|y\rangle \neq 0$, то значення степенів у доданках є різними, а значить коефіцієнт при $\left|y\right\rangle$ є нульовим. Відповідно, алгоритм Саймона

фактично зводить повний перебір станів регістру до розв'язання лінійного рівняння $\langle a|y\rangle = 0$, що можна зробити ефективно за час $O(n)$.

□

Класична трираундова схема Фейстеля у квантовій моделі

Виникає необхідність детально розглянути класичну трираундову схему Фейстеля у квантовій моделі обчислень, адже аналогічні твердження щодо її стійкості будуть сформовані у третьому розділі для шифру «Кипарис». Кувакадо Хіденорі та Мораї Масакату у статті [50] показали чому шифротекст

$$y = y_1 \parallel y_2 \in \{0,1\}^{2n}, \text{ де } y_1, y_2 \in \{0,1\}^n$$

схеми на рисунку 2.1 не є псевдовипадковою перестановкою у квантовій моделі обчислень, звівши питання щодо псевдовипадковості її виходу до задачі Саймона, коли для класичного випадку Майкл Лубі (англ. Michael Luby) та Чарльз Ракофф (англ. Charles Rackoff) у 1988 році довели, що трираундова схема Фейстеля є стійкою до атак розрізнення випадкового тексту від шифротексту [70].

На вхід до схеми на рисунку 2.1, яку позначимо як E, подається відкритий текст

$$x = x_1 \parallel x_2 \in \{0,1\}^{2n}, \text{ де } x_1, x_2 \in \{0,1\}^n.$$

Вважаємо, що P_1,P_2,P_3 — випадкові перестановки на n-бітових векторах, тоді на виході схеми шифрування отримаємо:

$$E(x) = E(x_1 \parallel x_2) = x_2 \oplus P_2(x_1 \oplus P_1(x_2)) \parallel$$
$$\parallel (x_1 \oplus P_1(x_2)) \oplus P_3(x_2 \oplus P_2(x_1 \oplus P_1(x_2))) = y_1 \parallel y_2.$$

Обернені перестановки до P_1,P_2,P_3 є невідомими. Очевидно, що для того, щоб відновити відкритий текст у цьому випадку, достатньо розглянути лише одну із частин шифротексту на третьому раунді, тобто або y_1, або y_2, адже обидві частини містять x_1 і x_2. У роботі [50] розглядали y_1, адже її вигляд не є таким складним. Для цього ввели позначення:

$$W(x) = W(x_1 \parallel x_2) = x_2 \oplus P_2(x_1 \oplus P_1(x_2)).$$

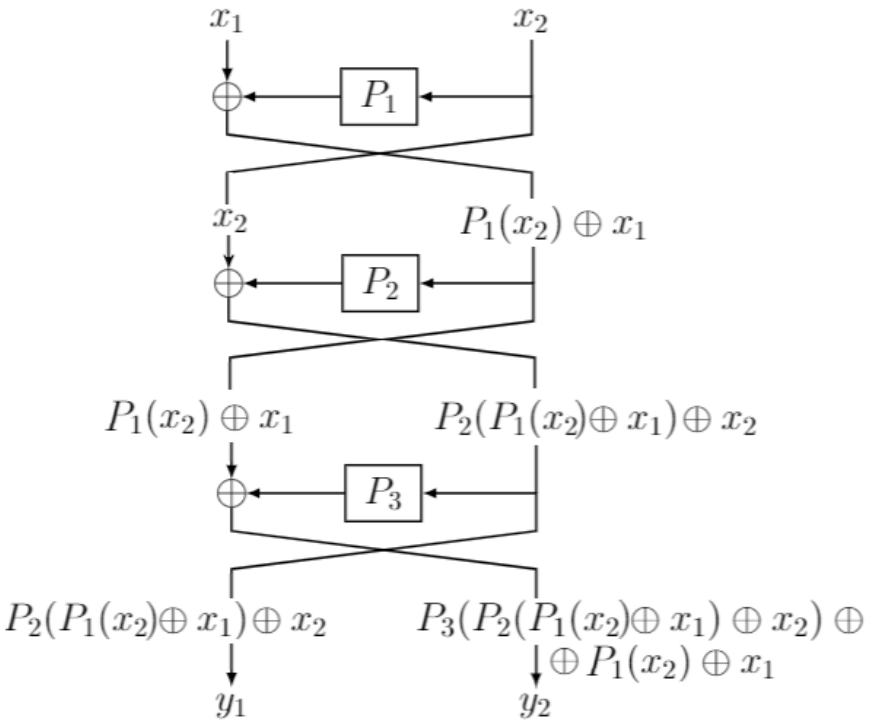


Рисунок 2.1 – Класична трираундова схема Фейстеля з розписаними виходами кожного раунду

Також побудували функцію f, яка приймає вектор із множини $(n+1)$-бітових векторів, а повертає вектор із множини n-бітових векторів. Функція f працює таким чином (α та β є деякими n-бітовими фіксованими векторами):

$$f(b \parallel x_1) = \begin{cases} W(x_1 \parallel \alpha) \oplus \beta, & \text{якщо } b = 0, \\ W(x_1 \parallel \beta) \oplus \alpha, & \text{якщо } b = 1. \end{cases}$$

Лема 2.1. *Вихідний шифротекст на третьому раунді класичної схеми Фейстеля (див. рис. 2.1) не є псевдовипадковою перестановкою у квантовій моделі обчислень.*

Доведемо цю лему, адже її ідея є підґрунтям квантової атаки на три раунди шифру «Кипарис», що буде представлена у третьому розділі.

Доведення. Покажемо, що $f(b \parallel x_1) = f(b' \parallel x_1')$ тоді та тільки тоді, коли $b' = b \oplus 1$ та $x_1' = x_1 \oplus a$, де період $a = P_1(\alpha) \oplus P_1(\beta)$.
Доведемо необхідність. Припустимо, що $b = b' = 0$, тоді:

$$f(0 \parallel x_1) = W(x_1 \parallel \alpha) \oplus \beta = \alpha \oplus P_2(x_1 \oplus P_1(\alpha)) \oplus \beta,$$
$$f(0 \parallel x_1^{'}) = W(x_1^{'} \parallel \alpha) \oplus \beta = \alpha \oplus P_2(x_1^{'} \oplus P_1(\alpha)) \oplus \beta.$$

Оскільки $f(0 \parallel x_1) = f(0 \parallel x_1^{'})$ та P_2 — випадкова перестановка, маємо:

$$x_1 = x_1^{'} \text{ та } b \parallel x_1 = b^{'} \parallel x_1^{'}.$$

Якщо $b = 0$, $b^{'} = 1$, тоді:

$$f(1 \parallel x_1^{'}) = W(x_1^{'} \parallel \beta) \oplus \alpha = \beta \oplus P_2(x_1^{'} \oplus P_1(\beta)) \oplus \alpha.$$

Оскільки $f(0 \parallel x_1) = f(1 \parallel x_1^{'})$ та P_2 — випадкова перестановка, маємо:

$$x_1^{'} = x_1 \oplus a, \text{ де період } a = P_1(\alpha) \oplus P_1(\beta).$$

Аналогічне доведення для випадків, коли $b = 1$, $b^{'} = 1$ та $b = 1$, $b^{'} = 0$.
Доведемо достатність. Нехай $b = 0, b^{'} = 1$, тоді:

$$f(1 \parallel x_1^{'}) = W(x_1^{'} \parallel \beta) \oplus \alpha = \beta \oplus P_2(x_1^{'} \oplus P_1(\beta)) \oplus \alpha =$$
$$= \beta \oplus P_2(x_1 \oplus P_1(\beta) \oplus P_1(\alpha) \oplus P_1(\beta)) \oplus \alpha = f(0 \parallel x_1).$$

Аналогічне доведення для випадку, коли $b = 1, b^{'} = 0$. □

Класична п'ятираундова схема Фейстеля у квантовій моделі

Через сім років після публікації роботи [50] Кувакадо Хіденорі та Морія Масакату у статті [54] Сяоян Донг (англ. Xiaoyang Dong) та Сяоюн Ван (англ. Xiaoyun Wang) показали як зробити атаку на класичну п'ятираундову схему Фейстеля у квантовій моделі обчислень, використовуючи комбінацію алгоритмів Саймона та Гровера (під алгоритмами маються на увазі конструктивні доведення відповідних задач). Опис задачі Саймона і її розв'язок вже був наведений вище. Розглянемо задачу Гровера [49], яку ще називають «задачею пошуку унікального елементу в невпорядкованій базі даних».

Задача Гровера

Нехай задана множина невпорядкованих елементів M, кожному з яких відповідає певне значення. Задача Гровера полягає у знаходженні деякого унікального елементу $x \in M$ по його значенню.

Очевидно, що у класичній моделі обчислень у найгіршому випадку для знаходження x треба перебрати всю множину M, яка може мати

експоненційну кількість елементів. У квантовій же моделі складність є у квадрат разів меншою [49, 51, 54, 71]. Перед тим як переходити до доведення цього факту, наведемо деякі міркування стосовно формулювання задачі Гровера, а також унітарних операторів, що використовуються у квантовому алгоритмі розв'язку задачі.

Якщо унікальним елементом вважати якийсь конкретний n-бітовий вектор x, то стає зрозуміло, що пошук у невпорядкованій базі даних за деяким n-бітовим значенням вектора є задачею пошуку прообразу деякої функції f на множині n-бітових векторів, знайшовши який, легко перевірити його коректність, підставивши у таку функцію (вектор v є шуканим значенням):

$$f_v(x) = \begin{cases} 1, & \text{якщо } v = x, \\ 0, & \text{якщо } v \neq x. \end{cases}$$

Таким чином, у квантовій моделі можна розглянути унітарний оператор U_{f_v}, який фактично є проєкцією оператора $U_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|y \oplus f(x)\right\rangle$ на підпростір $\left|x\right\rangle$, де $y = \frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}$, $\langle\cdot|\,, |\cdot\rangle$ — бра- та кет-вектори як у нотації Дірака:

$$U_{f_v} \left|x\right\rangle = (-1)^{f_v(x)} \left|x\right\rangle = \left|x\right\rangle - 2 \left|v\right\rangle \langle v|x\rangle.$$

Більш того, бачимо, що для вектора v оператор повертає $-\left|v\right\rangle$, адже скалярний добуток $\langle v|v\rangle = 1$, а для усіх інших базисних векторів підпростору — вектор $\left|x\right\rangle$, адже $\langle v|x\rangle = 0$.

Теорема 2.2. *Задача Гровера розв'язується у квантовій моделі обчислень за субекспоненційну кількість кроків [49, 51, 54, 71].*

Доведення.

1) Задамо n-кубітовий нульовий квантовий регістр $\left|0\right\rangle^n$, застосуємо перетворення Адамара до нього й отримаємо деякий стан $\left|l\right\rangle$:

$$\left|l\right\rangle = H_n \left|0\right\rangle^n = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} \left|x\right\rangle.$$

Якщо вважати, що шуканим n-бітовим вектором є $|v\rangle$, то геометрично розташування вектора $|l\rangle$ представляється як на рисунку 2.2а, де θ є кутом між $|l\rangle$ та гіперплощиною $2^n - 1$ векторів. Враховуючи те, що при малих значеннях кута θ, значення $\sin\theta$ також є малим, з рисунка маємо:

$$\theta = |\langle l|v\rangle| = \frac{1}{\sqrt{2^n}}.$$

2) Застосуємо оператор U_{f_v} до $|l\rangle$ і отримаємо відображення $|l\rangle$ відносно гіперплощини на кут θ. Результат виконання такої операції зображено на рисунку 2.2б.

3) За аналогією до оператора U_{f_v} визначимо U_{f_l} таким чином:

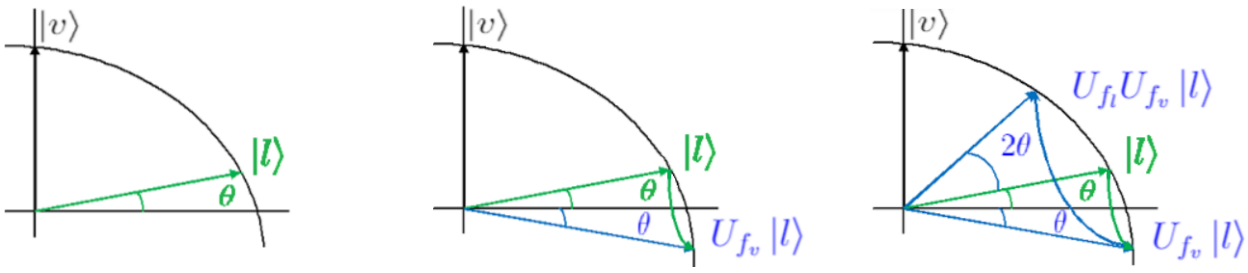
$$U_{f_l}|x\rangle = 2|l\rangle\langle l|x\rangle - |x\rangle.$$

Застосуємо його до $U_{f_v}|l\rangle$ і отримаємо відображення вектора $U_{f_v}|l\rangle$ на кут 2θ відносно вектора $|l\rangle$. Результат виконання такої операції зображено на рисунку 2.2в.

4) Оскільки вектор $|v\rangle$ є шуканим, то шляхом повторення кроків 2 та 3 дістаємось його. Під час кожного такого повторення отриманий вектор стає ближчим до $|v\rangle$ на 2θ, тому кількість кроків оцінюється як $\frac{\pi}{4\theta} - \frac{1}{2} \approx \frac{\pi\sqrt{2^n}}{4}$, тобто є субекспоненційною, що й треба було довести.

□

Зауваження. Часто у доведені теореми Гровера об'єднують кроки 2 та 3 і застосовують оператор $U_{f_l}U_{f_v}$. Отриманий крок називають *ітерацією Гровера*.



(а) **(б)** **(в)**

Рисунок 2.2 – Перші три кроки алгоритму Гровера

Тепер розглянемо п'ятираундову схему Фейстеля та атаку, що запропонована у роботі [54] Сяоян Донгом та Сяоюн Ваном. Нехай на вхід подається відкритий текст

$$x = x_1 \parallel x_2 \in \{0,1\}^n, \text{ де } x_1, x_2 \in \{0,1\}^{\frac{n}{2}},$$

а на виході маємо

$$y = y_1 \parallel y_2 \in \{0,1\}^n, \text{ де } y_1, y_2 \in \{0,1\}^{\frac{n}{2}}.$$

Вважаємо, що P_1,P_2,P_3,P_4,P_5 — випадкові перестановки на n-бітових векторах, що використовують n-бітові раундові ключі k_1,k_2,k_3,k_4,k_5 відповідно. З рисунка 2.3 бачимо, що якщо розписати перші три раунди шифру з початку та останні два з кінця, то з рівностей лівих частин між третім та четвертим раундами отримаємо:

$$P_2(P_1(x_2) \oplus x_1) = P_4(P_5(y_1) \oplus y_2) \oplus y_1 \oplus x_2.$$

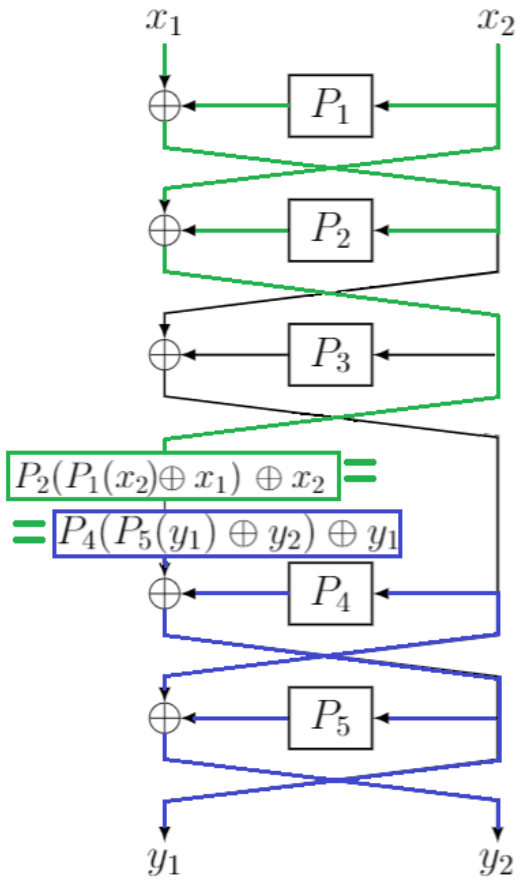


Рисунок 2.3 – Класична п'ятираундова схема Фейстеля

Визначимо функцію W таким чином:

$$W(y) = W(y_1 \parallel y_2) = P_4(P_5(y_1) \oplus y_2) \oplus y_1.$$

Згадуючи, описану вище атаку на класичну трираундову схему Фейстеля, де замість x_2 використовувались деякі вектори α або β, побудуємо функцію f, яка приймає вектор із множини $\frac{n}{2}+1$-бітових векторів, а повертає вектор із множини $\frac{n}{2}$-бітових векторів та має такий вигляд (α та β є деякими $\frac{n}{2}$-бітовими фіксованими векторами):

$$f(b \parallel y_1 \parallel y_2) = \begin{cases} W(y_1 \parallel y_2) \oplus \beta, & \text{якщо } b = 0, \\ W(y_1 \parallel y_2) \oplus \alpha, & \text{якщо } b = 1. \end{cases}$$

Тепер, якщо ключі k_4,k_5 знайдені коректно, то можемо вважати, що

$$P_2(P_1(x_2) \oplus x_1 \oplus a) = P_2(P_1(x_2) \oplus x_1),$$

де період $a = P_1(\alpha) \oplus P_1(\beta)$, інакше — ключі k_4,k_5 знайдені некоректно та вихід з $P_2(P_1(x_2) \oplus x_1)$ є псевдовипадковим. У статті [54] Сяоян Донг та Сяоюн Ван навели алгоритм, за яким фактично перебираючи усі значення k_4,k_5, можна знайти a.

Лема 2.2. *Атака відновлення ключів на четвертому та п'ятому раундах класичної схеми Фейстеля (див. рис. 2.3) у квантовій моделі обчислень є субекспоненційною [51, 54].*

Доведення. Задамо функцію $h(k_4,k_5,z_1,...,z_l) = f(z_1) \parallel ... \parallel f(z_l)$, де $z_i \in \{0,1\}^{\frac{n}{2}+1}$ — одна із можливих реалізацій $b \parallel y_1 \parallel y_2$, раундові ключі $k_4,k_5 \in \{0,1\}^{\frac{n}{2}}$. Тепер побудуємо квантового оракула U_h таким чином:

$$U_h \, |k_4,k_5,z_1,...,z_l,0,...,0\rangle = |k_4,k_5,z_1,...,z_l,h(k_4,k_5,z_1,...,z_l)\rangle.$$

Для того, щоб показати можливість знаходження періоду функції f, виконаємо такі кроки:

1) Задамо $(n+(\frac{n}{2}+1)l+\frac{n}{2}l)$-кубітовий нульовий регістр $|0\rangle^{n+(\frac{n}{2}+1)l+\frac{n}{2}l}$, застосуємо перетворення Адамара до перших $n+(\frac{n}{2}+1)l$ кубітів і отримаємо (значенням амплітуди знехтуємо):

$$\sum_{k_4,k_5=0}^{\sqrt{2^n}-1} \sum_{z_1,...,z_l=0}^{\sqrt{2^n}} |k_4,k_5\rangle\,|z_1\rangle\,...\,|z_l\rangle\,|0\rangle^{\frac{n}{2}l}.$$

2) Застосуємо оператор U_h до результату на минулому кроці:

$$\sum_{k_4,k_5=0}^{\sqrt{2^n}-1} \sum_{z_1,...,z_l=0}^{\sqrt{2^n}} |k_4,k_5\rangle\,|z_1\rangle\,...\,|z_l\rangle\,|h(k_4,k_5,z_1,...,z_l)\rangle.$$

3) Застосуємо перетворення Адамара до $|z_1\rangle$,..., $|z_l\rangle$ та отримаємо:

$$|\Phi\rangle = \sum_{k_4,k_5=0}^{\sqrt{2^n}-1} \sum_{\substack{u_1,...,u_l=0,\\ z_1,...,z_l=0}}^{\sqrt{2^n}} |k_4,k_5\rangle\,(-1)^{\langle u_1|z_1\rangle}\,|u_1\rangle\,...(-1)^{\langle u_l|z_l\rangle}\,|u_l\rangle\,|h(k_4,k_5,z_1,...,z_l)\rangle.$$

4) Виміримо стан регістру $|\Phi\rangle$ і, якщо значення k_4,k_5 є коректними, то період a, який розглядається як вектор, буде ортогональним до гіперплощини, що утворена множиною векторів u_1,...,u_l (вважатимемо, що $dim(u_1,...,u_l) = \frac{n}{2}$). Відповідно, застосуємо алгоритм Гровера, підставляючи у функцію f_v ключі k_4,k_5.

$$f_v(k_4,k_5) = \begin{cases} 1, & \text{якщо } k_4,k_5 \text{ коректні,} \\ 0, & \text{якщо } k_4,k_5 \text{ не є коректними.} \end{cases}$$

Через те, що ймовірність вибрати коректні значення ключів k_4,k_5 складає 2^{-n} (ключі є $\frac{n}{2}$-бітовими), то й складність застосування алгоритму Гровера у такому випадку є субекспоненційною.

□

Зауваження. У роботі [53] показано, що для знаходження періоду a достатньо вибрати $l = 2(\frac{n}{2} + 1 + \sqrt{\frac{n}{2} + 1})$.

Інші схеми Фейстеля у квантовій моделі

Не секрет, що існують різні узагальнення класичної схеми Фейстеля. Замість операції виключної диз'юнкції між раундами шифру можуть використовуватись побітове додавання, додавання за модулем та інші операції. Раундова функція може розміщуватись не тільки між так званими «гілками», які умовно ділять шифротекст на частини й вказують шлях, за яким йде кожна частина шифротексту, а й на самих гілках. Залежно від розташування раундової функції виділяють *L-схеми* (схеми

Misty), *R-схеми*, *B-схеми* (класичні схеми Фейстеля) *LR-схеми*, *LB-схеми*, *BR-схеми* та *LBR-схеми*. Також до схем Фейстеля відносять схеми, де відкритий текст розбивається не на дві частини, а на декілька. Їх представниками є узагальнені схеми Фейстеля типів I, II та III. Окрім цього, існують незбалансовані схеми, коли відкритий текст поділяють на частини різного розміру [72].

У 2017 році Сяоян Донг, Чжен Лі (англ. Zheng Li) та Сяоюн Ван показали як звести задачу розрізнення випадкового тексту від шифротексту для узагальнених схем Фейстеля типів I та II до задачі Саймона [51]. Для цього дослідники модифікували функції f та W, що вже введені вище, на випадок, коли відкритий текст поділяється не на дві частини, а на d частин, просто зробивши конкатенацію не двох n-бітових слів, а d. Розглянемо як це зробили на прикладі узагальненої схеми Фейстеля типу I (див. рис. 2.4). Нехай вхідним повідомленням до схеми є $4n$-бітовий вектор

$$x = x_1 \parallel x_2 \parallel x_3 \parallel x_4, \text{ де } x_1,x_2,x_3,x_4 \in \{0,1\}^n.$$

$P_1,...,P_7$ є випадковими перестановками на n-бітових векторах, обернені до яких вважаємо невідомим. Червоним кольором на рисунку 2.4 показан шлях, за яким розглядаємо вихід з семи раундів схеми, який позначимо як $W(x) = W(x_1 \parallel x_2 \parallel x_3 \parallel x_4) = P_4(P_3(P_2(P_1(x_4) \oplus x_3) \oplus x_2) \oplus x_1) \oplus x_4$. Функцію f, яка приймає $(4n+1)$-бітовий вектор, а повертає n-бітовий, задаємо таким чином (α та β є деякими n-бітовими векторами, x_1, x_2 — фіксовані):

$$f(b \parallel x_1 \parallel x_2 \parallel x_3) = \begin{cases} W(x_1 \parallel x_2 \parallel x_3 \parallel \alpha) \oplus \beta, & \text{якщо } b = 0, \\ W(x_1 \parallel x_2 \parallel x_3 \parallel \beta) \oplus \alpha, & \text{якщо } b = 1. \end{cases}$$

Лема 2.3. *Вихідний шифротекст на сьомому раунді узагальненої схеми Фейстеля типу I, де відкритий текст розбивається на чотири частини ($d = 4$, див. рис. 2.4) не є псевдовипадковою перестановкою у квантовій моделі обчислень.*

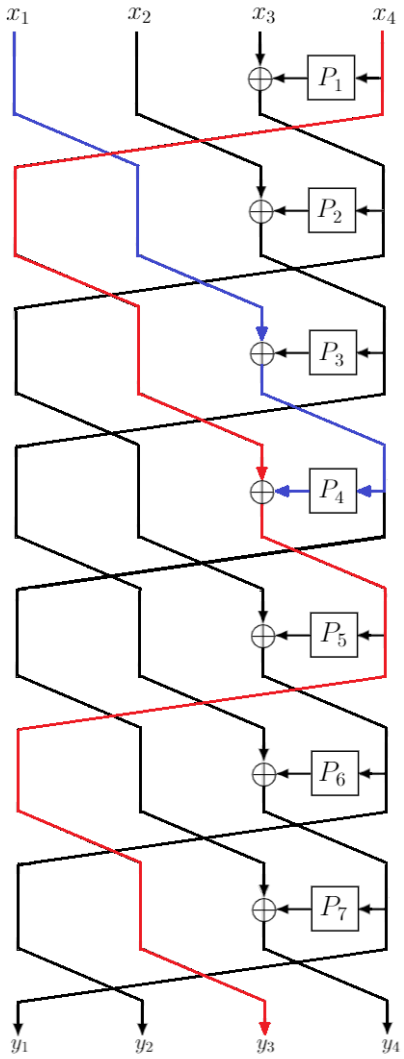


Рисунок 2.4 – Сім раундів узагальненої схеми Фейстеля типу I

Доведення. Покажемо, що $f(b \parallel x_1 \parallel x_2 \parallel x_3) = f(b' \parallel x_1 \parallel x_2 \parallel x_3')$ тоді та тільки тоді, коли $b' = b \oplus 1$ та $x_3' = x_3 \oplus a$, де період $a = P_1(\alpha) \oplus P_1(\beta)$. Доведемо необхідність. Припустимо, що $b = b' = 0$, тоді:

$$f(0 \parallel x_1 \parallel x_2 \parallel x_3) = W(x_1 \parallel x_2 \parallel x_3 \parallel \alpha) \oplus \beta =$$
$$= P_4(P_3(P_2(P_1(\alpha) \oplus x_3) \oplus x_2) \oplus x_1) \oplus \alpha \oplus \beta,$$
$$f(0 \parallel x_1 \parallel x_2 \parallel x_3') = W(x_1 \parallel x_2 \parallel x_3' \parallel \alpha) \oplus \beta =$$
$$= P_4(P_3(P_2(P_1(\alpha) \oplus x_3') \oplus x_2) \oplus x_1) \oplus \alpha \oplus \beta.$$

Оскільки $f(0 \parallel x_1 \parallel x_2 \parallel x_3) = f(0 \parallel x_1 \parallel x_2 \parallel x_3')$ та P_2 — випадкова перестановка, маємо:

$$x_3 = x_3' \text{ та } b \parallel x_1 \parallel x_2 \parallel x_3 = b' \parallel x_1 \parallel x_2 \parallel x_3'.$$

Якщо $b = 0$, $b^{'} = 1$, тоді

$$f(1 \parallel x_1 \parallel x_2 \parallel x_3^{'}) = W(x_1 \parallel x_2 \parallel x_3^{'} \parallel \beta) \oplus \alpha =$$
$$= P_4(P_3(P_2(P_1(\beta) \oplus x_3^{'}) \oplus x_2) \oplus x_1) \oplus \beta \oplus \alpha.$$

Оскільки $f(0 \parallel x_1 \parallel x_2 \parallel x_3) = f(1 \parallel x_1 \parallel x_2 \parallel x_3^{'})$ та P_2 — випадкова перестановка, маємо:

$$x_3^{'} = x_3 \oplus a, \text{ де період } a = P_1(\alpha) \oplus P_1(\beta).$$

Аналогічне доведення для випадків, коли $b = 1$, $b^{'} = 1$ та $b = 1, b^{'} = 0$.

Доведемо достатність. Нехай $b = 1, b^{'} = 0$, тоді:

$$f(1 \parallel x_1 \parallel x_2 \parallel x_3^{'}) = W(x_1 \parallel x_2 \parallel x_3^{'} \parallel \beta) \oplus \alpha =$$
$$= P_4(P_3(P_2(P_1(\beta) \oplus x_3^{'}) \oplus x_2) \oplus x_1) \oplus \beta \oplus \alpha =$$
$$= P_4(P_3(P_2(P_1(\beta) \oplus x_3 \oplus P_1(\alpha) \oplus P_1(\beta)) \oplus x_2) \oplus x_1) \oplus \beta \oplus \alpha =$$
$$= f(0 \parallel x_1 \parallel x_2 \parallel x_3).$$

Аналогічне можна довести для випадку, коли $b = 0$, $b^{'} = 1$. □

З доведення стає очевидно, що для d частин можемо атакувати $2d-1$ раундів узагальненої схеми Фейстеля типу I, якщо W задати як

$$W(x) = P_d(P_{d-1}(...P_2(P_1(x_d) \oplus x_{d-1}) \oplus x_{d-2})... \oplus x_2) \oplus x_1) \oplus x_d,$$

де відкритий текст

$$x = x_1 \parallel x_2 \parallel ... \parallel x_{d-2} \parallel x_{d-1} \parallel x_d \in \{0,1\}^{dn}, x_i \in \{0,1\}^n, i = \overline{1,d}.$$

Аналогічний результат Сяоян Донг, Чжен Лі та Сяоюн Ван сформулювали для узагальненої схеми Фейстеля типу II, для якої можна атакувати $2d + 1$ раундів. Однак загальний вигляд функції W не був наведен, що пояснюється його складністю.

2.2 Аналіз робіт з ротаційного криптоаналізу

Ймовірності, що обраховуються у R-криптоаналізі, є ймовірностями ротації для таких операцій як $\boxplus$, $\lll$,$\oplus$. Очевидно, що для операцій $\lll$ та

$\oplus$ така ймовірність дорівнює 1, а от з додаванням ситуація є складнішою, тому виникає необхідність у теоремі Магнуса Даума [19].

Теорема 2.3. *Нехай x та y є n-бітовими векторами, а r — фіксованим цілим числом з інтервалу $(0, n)$. Тоді:*
1. $Pr[(x \lll r_1) \lll r_2 = (x \lll r_2) \lll r_1] = 1$;
2. $Pr[x \lll r \oplus y \lll r = (x \oplus y) \lll r] = 1$;
3. $Pr[(x \boxplus y) \lll r = x \lll r \boxplus y \lll r] = \frac{1}{4}(1 + 2^{r-n} + 2^{-r} + 2^{-n})$.

З теореми 2.3 стає зрозуміло, що проблема зламу виникає коли у шифрі є додавання за модулем, а також зрозуміло, чому RX-схеми мають низьку складність. У своїй першій роботі за даною темою [20] Дмитро Ховратович і Івіца Ніколіч зробили припущення про незалежність послідовних додавань та сформулювали теорему про ймовірність зламу ARX-схеми.

Теорема 2.4. *Нехай q - кількість додавань за модулем в ARX-схемі з довільною кількістю операцій виключної диз'юнкції та циклічного зсуву. Тоді ймовірність ротації для ARX-схеми дорівнює $p_{\boxplus}^q$, де $p_{\boxplus}$ — ймовірність ротації для операції додавання за модулем, що залежить від параметра ротації r і вхідного слова довжини n.*

На жаль, припущення про незалежність послідовних додавань не виявилось дійсним, адже за принципом перемішування Шеннона для цього потрібна наявність операції, що розбиває два додавання, тому дослідники підрахували ймовірність для такого випадку [28]. Нижче наведену теорему надалі зручно називати «теоремою про послідовні додавання за модулем».

Теорема 2.5. *Нехай x_1,...,x_k є n-бітовими рівноймовірними векторами, а r — фіксованим цілим числом з інтервалу $(0, n)$. Тоді:*
$Pr[((x_1 \boxplus x_2) \lll r = x_1 \lll r \boxplus x_2 \lll r)$,
$[(x_1 \boxplus x_2 \boxplus x_3) \lll r = x_1 \lll r \boxplus x_2 \lll r \boxplus x_3 \lll r)$,...,
$((x_1 \boxplus ... \boxplus x_k) \lll r = x_1 \lll r \boxplus ... \boxplus x_k \lll r)] = \frac{1}{2^{nk}} \binom{k+2^r-1}{2^r-1} \binom{k+2^{n-r}-1}{2^{n-r}-1}$.

З таблиці 2.1 бачимо, що ймовірність того, що шифротекст відрізняється від випадкової перестановки менша при підрахунках за теоремою 2.5, ніж при обчисленні добутку ймовірностей з припущенням, що послідовність додавань за модулем утворює ланцюг Маркова. Також очевидно, що найбільшу ймовірність отримуємо тоді, коли r є маленьким, а найменшу — коли r дорівнює половині блоку вхідного повідомлення. Аналогічні результати отримані для лінійного [73] і диференціального криптоаналізів [74].

Таблиця 2.1 – Значення двійкових логарифмів ймовірностей ротації для послідовних додавань 64-бітових слів [28]

Параметр ротації $r = 1$								
Кількість додавань	1	2	3	4	5	6	7	8
за теоремою 2.4	-1.4	-2.8	-4.2	-5.7	-7.1	-8.5	-9.9	-11.3
за теоремою 2.5	-1.4	-3.6	-6.3	-9.3	-12.7	-16.3	-20.1	-24.1
Кількість додавань	9	10	11	12	13	14	15	16
за теоремою 2.4	-12.7	-14.1	-15.6	-17.0	-18.4	-19.8	-21.2	-22.6
за теоремою 2.5	-28.3	-32.7	-37.1	-41.7	-46.4	-51.3	-56.2	-61.2
Кількість додавань	17	18	19	20	21	22	23	24
за теоремою 2.4	-24.1	-25.5	-26.9	-28.3	-29.7	-31.1	-32.5	-34.0
за теоремою 2.5	-66.3	-71.4	-76.7	-82.0	-87.4	-92.9	-98.4	-104.0
Кількість додавань	25	26	27	28	29	30	31	32
за теоремою 2.4	-35.4	-36.8	-38.2	-39.6	-41.0	-42.4	-43.9	-45.3
за теоремою 2.5	-109.6	-115.3	-121.1	-126.9	-132.8	-138.7	-144.6	-150.6
Параметр ротації $r = 2$								
Кількість додавань	1	2	3	4	5	6	7	8
за теоремою 2.4	-1.7	-3.4	-5.0	-6.7	-8.4	-10.1	-11.7	-13.4
за теоремою 2.5	-1.7	-4.3	-7.5	-11.1	-15.1	-19.4	-23.9	-28.7
Кількість додавань	9	10	11	12	13	14	15	16
за теоремою 2.4	-15.1	-16.8	-18.4	-20.1	-21.8	-23.5	-25.1	-26.8
за теоремою 2.5	-33.6	-38.7	-44.0	-49.4	-54.9	-60.6	-66.3	-72.2
Кількість додавань	17	18	19	20	21	22	23	24
за теоремою 2.4	-28.5	-30.2	-31.8	-33.5	-35.2	-36.9	-38.5	-40.2
за теоремою 2.5	-78.1	-84.2	-90.3	-96.5	-102.8	-109.1	-115.5	-122.0
Кількість додавань	25	26	27	28	29	30	31	32
за теоремою 2.4	-41.9	-43.6	-45.3	-46.9	-48.6	-50.3	-52.0	-53.6
за теоремою 2.5	-128.5	-135.1	-141.8	-148.5	-155.3	-162.1	-169.0	-175.9

Оскільки у підрозділі 3.3 для ротаційного криптоаналізу шифру «Кипарис» будуть використані теореми 2.4 та 2.5, то спочатку наведемо приклад їх застосування до геш-функції BLAKE2.

Приклад 2.1. У роботі [20] Дмитро Ховратович і Івіца Ніколіч обчислили ймовірність ротації функції перестановки P у BLAKE2. Розглянемо як саме це зробили. Спочатку треба обчислити ймовірності послідовних додавань за модулем за теоремою 2.5, а потім взяти добуток таких ймовірностей як у теоремі 2.4. На рисунку 1.1 наведена візуалізація перестановки P, на якій червоним виділено 8 «ланцюгів» з чотирма послідовними додаваннями за модулем. Для збільшення ймовірності за шифротекстом сказати вхід, автори розглядали пари зсуву на 1. Також вони вважали, що між раундами немає нічого, що об'єднувало усі 8 ланцюгів чи робило певне перетворення результатів додавань.

Отже, якщо розглянемо 8 раундів шифрування, то, у такому випадку, матимемо 8 незалежних ланцюгів з $4 \cdot 8$ додаваннями кожен і, використовуючи таблицю 2.1, отримаємо ймовірність

$$(2^{-150.6})^8 = 2^{-1204.8},$$

коли ймовірність випадкової перестановки становить 2^{-1024}. Таким чином, P не є вразливою до R-криптоаналізу, адже містить 12 раундів.

Означення 2.3. Константа c називається *інваріантною до циклічного зсуву на* γ, якщо $c = c \lll \gamma$ для деякого фіксованого значення γ.

Очевидно, що якщо до пари ротації на γ застосувати операцію виключної диз'юнкції з інваріантною до циклічного зсуву на γ константою, то за теоремою 2.3 ймовірність ротації не зміниться, коли такі ж перетворення з неінваріантною константою змінюють ймовірність, адже тоді згадана теорема не може застосовуватись. Отримана пара перестає бути парою ротації та потребує детальнішого дослідження. Відповідно, з'явились такі поняття як RX-пара та RX-різниця (про які вже згадувалось у підрозділі 1.2).

Варто зазначити, що RX-пара є R-парою, якщо $a_1 = a_2 = 0$.

Означення 2.4. *RX-характеристикою* криптографічного примітива називають послідовність усіх RX-різниць від входу у перший раунд шифрування до виходу з останнього раунду шифрування.

Томер Ашур та Янвен Лю хотіли оцінити ймовірність збереження RX-різниці після проходження певної кількості раундів шифрування. Вони припустили, що мають $(x \oplus a_1, \overleftarrow{x} \oplus a_2)$ та $(y \oplus b_1, \overleftarrow{y} \oplus b_2)$ RX-пари та сформулювали теорему 2.6.

Теорема 2.6. *Нехай x,y є n-бітовими незалежними рівномірно розподіленими векторами, а a_1,b_1,a_2,b_2,Δ_1,Δ_2 — фіксованими цілими числами з інтервалу (0,n). Тоді:*

$$Pr[\overleftarrow{(x \oplus a_1) \boxplus (y \oplus b_1) \oplus \Delta_1} = (\overleftarrow{x} \oplus a_2) \boxplus (\overleftarrow{y} \oplus b_2) \oplus \Delta_2] =$$
$$= 1_{(I \oplus SHL)(\delta_1 \oplus \delta_2 \oplus \delta_3) \oplus 1 \preceq SHL((\delta_1 \oplus \delta_3)|(\delta_2 \oplus \delta_3))} \cdot 2^{-|SHL((\delta_1 \oplus \delta_3)|(\delta_2 \oplus \delta_3))|} \cdot 2^{-3} +$$
$$+ 1_{(I \oplus SHL)(\delta_1 \oplus \delta_2 \oplus \delta_3) \preceq SHL((\delta_1 \oplus \delta_3)|(\delta_2 \oplus \delta_3))} \cdot 2^{-|SHL((\delta_1 \oplus \delta_3)|(\delta_2 \oplus \delta_3))|} \cdot 2^{-1.415},$$
$$\text{де } \delta_1 = R(a_1) \oplus L'(a_2), \quad \delta_2 = R(b_1) \oplus L'(b_2), \quad \delta_3 = R(\Delta_1) \oplus L'(\Delta_2),$$

$\overleftarrow{x} = x \lll 1$,

$R(x)$ *є вектором* $n - \gamma$ *наймолодших значущих бітів вектора* x *(*γ *деяке число, що менше за* n *і поділяє* x *на дві частини),*

$L'(x)$ *—* $n - \gamma$ *найстарших значущих бітів вектора* x*.*

Зауваження. Третій пункт теореми 2.3 є частковим випадком теореми 2.6, коли $a_1 = a_2 = b_1 = b_2 = \Delta_1 = \Delta_2 = 0$.

Окрім математичного обґрунтування, яке підтверджує коректність даної теореми, дослідники реалізували жадібний алгоритм, що робить прямі обрахунки, та впевнились у своїх результатах на прикладі блокового шифру SPECK [31].

Приклад 2.2. У статті [31] розглядали версію SPECK32/64 ($n = 16, m = 4, \alpha = 7, \beta = 2$). Спочатку встановили початкові константи a_1,a_2,b_1,b_2,Δ_1,Δ_2 рівними нулю та підрахували ймовірності окремо для кожного з 6 раундів шифрування, використовуючи теорему 2.6 та

програмну реалізацію написаного жадібного алгоритму для усіх можливих пар (x, y). Результати виявились однаковими. Тобто у роботі [31] практично показали справедливість теореми. Однак розглядати результати на окремих раундах шифрування не є гарною практикою з криптографічної точки зору, адже нас цікавить вихідна ймовірність усієї криптографічної схеми, тому Томер Ашур та Янвен Лю спробували вважати раунди незалежними та, відповідно, обчислювати ймовірність виходу на 6 раунді SPECK як добуток ймовірностей на кожному раунді. На жаль, обчислювальні можливості дослідників були обмеженими, тому вони робили обрахунки лише для 2^{33} випадково вибраних ключів, а потім отримані результати порівнювали з практичними. Вихідна ймовірність програми виявилось нижчою за добуток ймовірностей на кожному раунді шифрування, що вказувало на залежність $x^{(i)}$ та $y^{(i)}$, а також на існування слабого класу ключів. За їх обрахунками потужність цього класу складала 2^{39}.

На 6-ти раундах шифрування дослідники не зупинились, а вирішили діяти таким чином: спочатку згенерувати ключ шифрування, перевірити його на слабкість (порівнюючи з отриманими результатами на 6-му раунді), а потім використати його для шифрування усіх можливих пар (x, y) на 7-ми раундах і порівняти результати підрахунків їх програми з добутком ймовірностей на кожному раунді шифрування за теоремою 2.6. Початкові константи a_1,a_2,b_1,b_2,Δ_1,Δ_2 вони знову встановили рівними нулю. Виявилось, що з 2^{64} випадково обраних ключів, кількість слабких ключів складала 2^7, коли на 6 раунді шифрування вони оцінювали їх кількість як 2^{39}. Найцікавішим є те, що результати підрахунків програми на слабких ключах майже збігались з добутком ймовірностей на кожному з 7 раундів шифрування за теоремою 2.6, а ймовірність того, що RX-різниця ((0,0),1) зберігатиметься усі 7 раундів дорівнювала $2^{-31.6}$, коли для випадкової перестановки така ймовірність складає

$$(2^{-32})^7 = 2^{-224}.$$

Зауваження. Томер Ашур та Янвен Лю не підрахували скільки

раундів шифрування у SPECK32/64 потрібно для повної протидії RX-криптоаналізу, але дали над чим подумати.

Зауваження. Було зроблено припущення стосовно ключового розподілу: на кожному раунді шифрування розглядався однаковий ключ.

Зауваження. У проведеному криптоаналізі SPECK32/64 зроблено негласне припущення щодо ймовірності ротації правої частини. За даними, що наведені у статті [31], ймовірність ротації правої частини y_i, де i — номер раунду, приймають за одиницю.

Зауваження. Наведений приклад говорить про те, що добре б поширити теорему 2.6 на випадок, коли є послідовні додавання за модулем.

Зауваження. У 2016 році Лінг Сонг (Ling Song) і його колеги зробили найкращу на сьогодні атаку на SPECK. Вони показали, що для 64-бітового блоку та 128-бітового ключа диференціальний криптоаналіз працює для 20 раундів шифрування. Однак за стандартами шифр SPECK при такому розмірі блоку та ключа має 27 раундів, тому він є стійким до диференціального криптоаналізу [75, 76].

У роботі 2020 року [35] застосовано ротаційний криптоаналіз до функції перестановки шифру ChaCha [13], сформована теорема 2.7 та наслідок 2.1.

Теорема 2.7. *Нехай x_1,...,x_k є n-бітовими рівноймовірними та незалежними векторами, а r — фіксованим цілим числом з інтервалу $(0, n)$. Тоді:*

$$Pr[\boxplus_{i=1}^{k}(x_i \lll r) = (\boxplus_{i=1}^{k} x_i) \lll r] = \frac{F(r,k,n)\cdot F(n-r,k,n)}{2^{kn}},$$

$$\text{де } F(r,k,n) = \sum_{h=0}^{\lfloor \frac{k(2^r-1)}{2^n} \rfloor} \sum_{j=0}^{k} (-1)^j \binom{k}{j} \left(\binom{h2^n-(j-1)2^r-1+k}{h2^n-(j-1)2^r-1} - \binom{h2^n-j2^r-1+k}{h2^n-j2^r-1} \right).$$

Наслідок 2.1. *Нехай x, y, z є n-бітовими рівноймовірними та незалежними векторами, а r — фіксованим цілим числом з інтервалу $(0, n)$. Тоді:*

$$Pr[(x \lll r) \boxplus (y \lll r) \boxplus (z \lll r) = (x \boxplus y \boxplus z) \lll r] =$$
$$= \frac{D(2^r+2)(2^{n-r}+2)}{9 \cdot 2^n} + 1_{\{r=1 \vee r=n-1\}} \frac{4}{2^{3n}} \binom{2^n-1}{2^{n-1}-3},$$

де D — ймовірність, що визначається третім пунктом теореми 2.3.

Зауваження. Якщо у наслідку 2.1 обрати r з інтервалу $(1, n-1)$, то отриманий результат збігатиметься з результатом теореми 2.5 при $k = 3$, а якщо у наслідку 2.1 обрати $r = 1$ чи $r = n-1$, то ймовірність буде більшою за отриману у теоремі 2.5.

Висновки до розділу 2

У даному розділі розглядався квантовий криптоаналіз фейстель-подібних схем шифрування. Для цього описані основні результати наявних робіт, зокрема стаття Кувакадо Хіденорі та Морія Масакату, у якій дослідники, застосовуючи алгоритм Саймона, показали квантову нестійкість класичної трираундової схеми Фейстеля; стаття Сяоян Донга та Сяоюн Вана, у якій представлена атака вже на п'ятираундову схему, для проведення якої використовували комбінацію алгоритмів Саймона та Гровера. Квантових атак, що не є атаками повного перебору, на більшу кількість раундів на цей момент невідомо, але зазначені будуть використані для проведення квантового криптоаналізу малоресурсного симетричного блокового шифру «Кипарис» у третьому розділі даної роботи.

Окрім цього, доведена вразливість узагальненої схеми Фейстеля типу I у квантовій моделі обчислень, для успішної атаки розрізнення випадкового тексту від шифротексту $2d - 1$ раундів якої необхідно та достатньо, щоб відкритий текст на вході поділявся на d частин. За аналогією до цієї атаки у третьому розділі буде зроблена атака на узагальнену схему Фейстеля типу III.

Також наведені наявні результати з ротаційного криптоаналізу ARX-шифрів, який був відомий ще у минулому столітті, але тільки після

виходу статті Дмитра Ховратовіч та Івіци Ніколіча 2015 року став набирати все більшої популярності. Розглянуті роботи стосовно ротаційного криптоаналізу з введеними через операцію виключної диз'юнкції константами у вхідне повідомлення. У якості прикладів застосування таких криптоаналізів у цьому розділі представлені атаки на геш-функцію BLAKE2 та на малоресурсний симетричний блоковий шифр SPECK32/64, за аналогією до яких у третьому розділі проведений ротаційний криптоаналіз шифру «Кипарис».

3 КРИПТОАНАЛІЗ ШИФРУ «КИПАРИС»

Обґрунтування стійкості будь-якого шифру є доволі об'ємною і непростою задачею, адже існує багато методів та підходів до його криптоаналізу. У роботі досліджується малоресурсний симетричний блоковий шифр «Кипарис», який у 2017 році Марія Родінко та Роман Олійніков представили у статті [15]. Відомо, що він є практично стійким до диференціального криптоаналізу [60, 77]. У цьому розділі вперше проведено дослідження стійкості шифру «Кипарис» до квантового та ротаційного криптоаналізів, а також доведена вразливість узагальненої схеми Фейстеля типу III у квантовій моделі обчислень.

3.1 Застосування квантового криптоаналізу до шифру «Кипарис»

Покажемо, що трираундова схема шифру «Кипарис» не є стійкою до атак розрізнення випадкового тексту від шифротексту у квантовій моделі обчислень. Для цього застосуємо ідеї Кувакадо Хіденорі та Морія Масакату, що представлені у роботі [50], звівши питання щодо випадковості шифротексту до задачі Саймона. З рисунка 1.3 бачимо, що відкритим n-бітовим текстом є x (n вважаємо парним числом), а раундовою функцією — F, у якій спочатку відбувається додавання з раундовим ключем, а потім двократне застосування ARX-функції h до половини блоку. Відкритий текст x та вихідний текст з трьох раундів шифрування y позначимо як конкатенацію лівої та правої частин:

$$x = L_0 \parallel R_0,\ y = L_3 \parallel R_3, \text{ де } L_0,R_0,L_3,R_3 \in \{0,1\}^{\frac{n}{2}}.$$

Для відновлення відкритого тексту x розглядаємо лише праву частину

$$R_3 = L_0 \oplus F_{RK^{(2)}}(R_0 \oplus F_{RK^{(1)}}(L_0)),$$

де $F_{RK^{(1)}}, F_{RK^{(2)}}$ — раундові функції з ключами $RK^{(1)}$ та $RK^{(2)}$ відповідно. Позначимо

$$W(x) = W(L_0 \parallel R_0) = L_0 \oplus F_{RK^{(2)}}(R_0 \oplus F_{RK^{(1)}}(L_0)) = L_0 \oplus F_{RK^{(2)}}(R_0 \oplus \\ \oplus h^2(L_0 \oplus RK^{(1)})) = L_0 \oplus h^2(R_0 \oplus h^2(L_0 \oplus RK^{(1)}) \oplus RK^{(2)}).$$

Задамо функцію f таким чином:

$$f(b \parallel R_0) = \begin{cases} W(\alpha \parallel R_0) \oplus \beta, & \text{якщо } b = 0, \\ W(\beta \parallel R_0) \oplus \alpha, & \text{якщо } b = 1. \end{cases}$$

Для того, щоб розписати f для трьох раундів шифру «Кипарис», користуємось ARX-функцією h від деякого значення $a = a_0 \parallel a_1 \parallel a_2 \parallel a_3$, вихід з якої має вигляд:

$$h(a) = h(a_0 \parallel a_1 \parallel a_2 \parallel a_3) = a_h^0 \parallel a_h^1 \parallel a_h^2 \parallel a_h^3.$$

Виконуючи просте розгортання з кінця вихідного тексту функції h (див. рис. 1.3), отримаємо конкатенацію таких значень:
$a_h^0 = a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)$;
$a_h^1 = (((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2) \oplus ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll \\ \lll r_1)) \oplus (((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll \\ \lll r_1))) \lll r_2)) \lll r_3)) \lll r_4$;
$a_h^2 = a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1)) \oplus ((((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus \\ \boxplus((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2))) \lll r_3$
$a_h^3 = (((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll \\ \lll r_1))) \lll r_2))) \lll r_3$.

Підставивши у цю ж функцію замість a її значення $h(a)$, отримаємо:

$$h^2(a) = h(a_h^0 \parallel a_h^1 \parallel a_h^2 \parallel a_h^3) = a_{h^2}^0 \parallel a_{h^2}^1 \parallel a_{h^2}^2 \parallel a_{h^2}^3.$$

Значення $a_{h^2}^0$,$a_{h^2}^1$,$a_{h^2}^2$,$a_{h^2}^3$ наведені у додатку А.

Тобто, якщо у формулі для $h^2(a)$ замість a підставити $L_0 \oplus RK^{(1)}$, а потім зробити ще один виклик такої функції вже від

$$R_0 \oplus h^2(L_0 \oplus RK^{(1)}) \oplus RK^{(2)}$$

і до отриманого результату застосувати операцію виключної диз'юнкції з L_0, то отримаємо розписану функцію f для трьох раундів шифру «Кипарис».

Через те, що вихідна формула є дуже великою, вона не наводиться у роботі, але наведемо лему аналогічну лемі 2.1, де α та β вважаємо деякими $\frac{n}{2}$-бітовими фіксованими векторами, $RK^{(1)}$,$RK^{(2)}$ — фіксованими значеннями ключів, а h^2 — випадковою перестановкою.

Твердження 3.1. *Вихідний шифротекст на третьому раунді малоресурсного симетричного шифру «Кипарис» не є псевдовипадковою перестановкою у квантовій моделі обчислень.*

Доведення. Зведемо питання щодо випадковості вихідного шифротексту з трьох раундів схеми «Кипарис» до задачі Саймона. Для цього покажемо, що $f(b \parallel R_0) = f(b \parallel R_0^{'})$ тоді та тільки тоді, коли $b^{'} = b \oplus 1$ та $R_0^{'} = R_0 \oplus a$, де період $a = h^2(\alpha \oplus RK^{(1)}) \oplus h^2(\beta \oplus RK^{(1)})$. Доведемо необхідність. Припустимо, що $b = b^{'} = 0$, тоді:

$$f(0 \parallel R_0) = W(\alpha \parallel R_0) \oplus \beta = \alpha \oplus h^2(R_0 \oplus h^2(\alpha \oplus RK^{(1)}) \oplus RK^{(2)}) \oplus \beta,$$
$$f(0 \parallel R_0^{'}) = W(\alpha \parallel R_0^{'}) \oplus \beta = \alpha \oplus h^2(R_0^{'} \oplus h^2(\alpha \oplus RK^{(1)}) \oplus RK^{(2)}) \oplus \beta.$$

Оскільки $f(0 \parallel R_0) = f(0 \parallel R_0^{'})$ та h^2 — випадкова перестановка, маємо:

$$R_0 = R_0^{'} \text{ та } b \parallel R_0 = b^{'} \parallel R_0^{'}.$$

Якщо $b = 0$, $b^{'} = 1$, тоді:

$$f(1 \parallel R_0^{'}) = W(\beta \parallel R_0^{'}) \oplus \alpha = \beta \oplus h^2(R_0^{'} \oplus h^2(\beta \oplus RK^{(1)}) \oplus RK^{(2)}) \oplus \alpha.$$

Оскільки $f(0 \parallel R_0) = f(1 \parallel R_0^{'})$ та h^2 — випадкова перестановка, маємо:

$$R_0^{'} = R_0 \oplus a, \text{ де період } a = h^2(\alpha \oplus RK^{(1)}) \oplus h^2(\beta \oplus RK^{(1)}).$$

Аналогічне доведення для випадків, коли $b = 1$, $b^{'} = 1$ та $b = 1$, $b^{'} = 0$. Доведемо достатність. Нехай $b = 1, b^{'} = 0$, тоді:

$$f(1 \parallel R_0^{'}) = W(\beta \parallel R_0^{'}) \oplus \alpha = \beta \oplus h^2(R_0^{'} \oplus h^2(\beta \oplus RK^{(1)}) \oplus RK^{(2)}) \oplus \alpha =$$
$$= \beta \oplus h^2(R_0 \oplus h^2(\alpha \oplus RK^{(1)}) \oplus h^2(\beta \oplus RK^{(1)}) \oplus h^2(\beta \oplus RK^{(1)}) \oplus RK^{(2)}) \oplus \alpha =$$
$$= f(0 \parallel R_1).$$

Аналогічне можна довести для випадку, коли $b = 0, b' = 1$. □

Через те, що доведення існування поліноміального розв'язку задачі Саймона у квантовій моделі обчислень (див. доведення теореми 2.1) є конструктивним, то підставивши замість абстрактної функції f у нього задану вище, отримаємо алгоритм знаходження періоду a для трьох раундів шифру «Кипарис».

Окрім цього, можна дослідити стійкість більшої кількості раундів «Кипарис», наприклад, спочатку застосувати алгоритм Саймона для трьох раундів шифру, а потім, використовуючи алгоритм Гровера, провести атаку відновлення ключів на наступних раундах за аналогією до роботи [54] Сяоян Донга та Сяоюн Вана для класичної схеми Фейстеля. Для доведення твердження 3.2 вважатимемо, що відкритим текстом є

$$x = L_0 \parallel R_0 \in \{0,1\}^n, \text{ де } L_0, R_0 \in \{0,1\}^{\frac{n}{2}}.$$

Відповідно, вихід з i-того раунду шифрування позначатимемо так:

$$L_i = R_{i-1} \oplus F(L_{i-1}, RK^{(i-1)}),\ R_i = L_{i-1}.$$

Твердження 3.2. *Атака відновлення ключів на четвертому та п'ятому раундах шифру «Кипарис» у квантовій моделі обчислень має субекспоненційну складність.*

Доведення. Для проведення атаки розглянемо п'ятираундову схему шифрування «Кипарис» та розпишемо перші три раунди з початку та останні два з кінця, тоді з рівностей правих частин між третім та четвертим раундами матимемо:

$$F_{RK^{(2)}}(R_0 \oplus F_{RK^{(1)}}(L_0)) = R_5 \oplus F_{RK^{(4)}}(L_5 \oplus F_{RK^{(5)}}(R_5)) \oplus L_0.$$

Визначимо функцію f, яка приймає вектор із множини $\frac{n}{2} + 1$-бітових векторів, а повертає вектор із множини $\frac{n}{2}$-бітових векторів таким чином (α та β є деякими $\frac{n}{2}$-бітовими фіксованими векторами):

$$f(b \parallel L_5 \parallel R_5) = \begin{cases} R_5 \oplus F_{RK^{(4)}}(L_5 \oplus F_{RK^{(5)}}(R_5)) \oplus \beta, & \text{якщо } b = 0, \\ R_5 \oplus F_{RK^{(4)}}(L_5 \oplus F_{RK^{(5)}}(R_5)) \oplus \alpha, & \text{якщо } b = 1. \end{cases}$$

Тепер, якщо ключі $RK^{(4)}$,$RK^{(5)}$ знайдені коректно, то матимемо

$$F_{RK^{(2)}}(R_0 \oplus a \oplus F_{RK^{(1)}}(L_0)) = F_{RK^{(2)}}(R_0 \oplus F_{RK^{(1)}}(L_0)),$$

де період

$$a = h^2(\alpha \oplus RK^{(1)}) \oplus h^2(\beta \oplus RK^{(1)})$$

знаємо з твердження 3.1, інакше — ключі $RK^{(4)}$,$RK^{(5)}$ знайдені некоректно та вихід з $F_{RK^{(2)}}(R_0 \oplus F_{RK^{(1)}}(L_0))$ є псевдовипадковим.

Задамо функцію

$$h(RK^{(4)},RK^{(5)},z_1,...,z_l) = f(z_1) \parallel ... \parallel f(z_l),$$

де $z_i \in \{0,1\}^{\frac{n}{2}+1}$ — одна із можливих реалізацій $b \parallel L_5 \parallel R_5$, раундові ключі $RK^{(4)}$,$RK^{(5)} \in \{0,1\}^{\frac{n}{2}}$. Тепер побудуємо квантового оракула U_h:

$$U_h \left|RK^{(4)},RK^{(5)},z_1,...,z_l,0,...,0\right\rangle =$$
$$= \left|RK^{(4)},RK^{(5)},z_1,...,z_l,h(RK^{(4)},RK^{(5)},z_1,...,z_l)\right\rangle.$$

Для того, щоб показати можливість знаходження періоду функції f, виконаємо такі кроки:

1) Задамо $(n+(\frac{n}{2}+1)l+\frac{n}{2}l)$-кубітовий нульовий регістр $|0\rangle^{n+(\frac{n}{2}+1)l+\frac{n}{2}l}$, застосуємо перетворення Адамара до перших $n+(\frac{n}{2}+1)l$ кубітів і отримаємо (значенням амплітуди знехтуємо):

$$\sum_{RK^{(4)},RK^{(5)}=0}^{\sqrt{2^n}-1} \sum_{z_1,...,z_l=0}^{\sqrt{2^n}} \left|RK^{(4)},RK^{(5)}\right\rangle \left|z_1\right\rangle ... \left|z_l\right\rangle |0\rangle^{\frac{n}{2}l}.$$

2) Застосуємо оператор U_h до результату на минулому кроці:

$$\sum_{RK^{(4)},RK^{(5)}=0}^{\sqrt{2^n}-1} \sum_{z_1,...,z_l=0}^{\sqrt{2^n}} \left|RK^{(4)},RK^{(5)}\right\rangle \left|z_1\right\rangle ... \left|z_l\right\rangle \left|h(RK^{(4)},RK^{(5)},z_1,...,z_l)\right\rangle.$$

3) Застосуємо перетворення Адамара до $\left|z_1\right\rangle$,..., $\left|z_l\right\rangle$ та отримаємо:

$$|\Phi\rangle = \sum_{RK^{(4)},RK^{(5)}=0}^{\sqrt{2^n}-1} \sum_{\substack{u_1,...,u_l=0,\\ z_1,...,z_l=0}}^{\sqrt{2^n}} \left|RK^{(4)},RK^{(5)}\right\rangle (-1)^{\langle u_1|z_1\rangle} \left|u_1\right\rangle ...$$
$$...(-1)^{\langle u_l|z_l\rangle} \left|u_l\right\rangle \left|h(RK^{(4)},RK^{(5)},z_1,...,z_l)\right\rangle$$

4) Виміримо стан регістру $|\Phi\rangle$ і, якщо значення $RK^{(4)}$,$RK^{(5)}$ є коректними, то період a, який розглядається як вектор, буде

ортогональним до гіперплощини, що утворена множиною векторів u_1,...,u_l (вважатимемо, що $dim(u_1,...,u_l) = \frac{n}{2}$). Відповідно, застосуємо алгоритм Гровера (див. доведення теореми 2.2). Функція f_v виглядатиме таким чином:

$$f_v(RK^{(4)},RK^{(5)}) = \begin{cases} 1, & \text{якщо } RK^{(4)},RK^{(5)} \text{ коректні,} \\ 0, & \text{якщо } RK^{(4)},RK^{(5)} \text{ не є коректними.} \end{cases}$$

Через те, що ймовірність вибрати коректні значення ключів $RK^{(4)}$,$RK^{(5)}$ складає 2^{-n} (ключі є $\frac{n}{2}$-бітовими), то й складність застосування алгоритму Гровера у такому випадку є субекспоненційною.

□

Наслідок 3.1. *Для версії «Кипарис-256» ймовірність проведення атаки відновлення ключів на четвертому та п'ятому раундах складає*

$$2^{-\frac{256}{2}} = 2^{-128},$$

а для версії «Кипарис-512» — $2^{-\frac{512}{2}} = 2^{-256}$.

Наслідок 3.2. *Для атаки на і раундів шифрування «Кипарис» треба перебрати* $RK^{(4)}$,...,$RK^{(i)}$ *ключів, що, використовуючи алгоритм Гровера, можна зробити з ймовірністю* $2^{-\frac{n}{2}\cdot i}$.

Наслідок 3.3. *Ймовірність проведення атаки відновлення ключів* $RK^{(4)}$,...,$RK^{(10)}$ *схеми шифрування «Кипарис-256» складає* 2^{-1280}, *а для відновлення* $RK^{(4)}$,...,$RK^{(14)}$ *«Кипарис-512» —* 2^{-3584} *у квантовій моделі обчислень.*

3.2 Узагальнена схема Фейстеля типу III у квантовій моделі

Сяоян Донг, Чжен Лі та Сяоюн Ван у роботі [51] запропонували шлях зведення задачі розрізнення випадкового тексту від шифротексту для узагальнених схем Фейстеля типів I та II до задачі Саймона. Аналогічний результат можна показати і для типу III.

Розглянемо схему на рисунку 3.1 і прослідкуємо виходи з кожного її раунду. Нехай вхідним повідомленням до даної схеми є $3n$-бітовий вектор

$$x = x_1 \parallel x_2 \parallel x_3, \text{ де } x_1,x_2,x_3 \in \{0,1\}^n.$$

Функції перестановки позначимо P_i^j, де i — номер раунду, j — номер функції перестановки на раунді (вважаємо, що нумерація починається з 1). З рисунку 3.1 видно, що другий блок на третьому раунді має вигляд:

$$W(x) = W(x_1 \parallel x_2 \parallel x_3) = P_3^1(P_2^1(P_1^1(x_2) \oplus x_1) \oplus x_3) \oplus P_1^2(x_3) \oplus x_2.$$

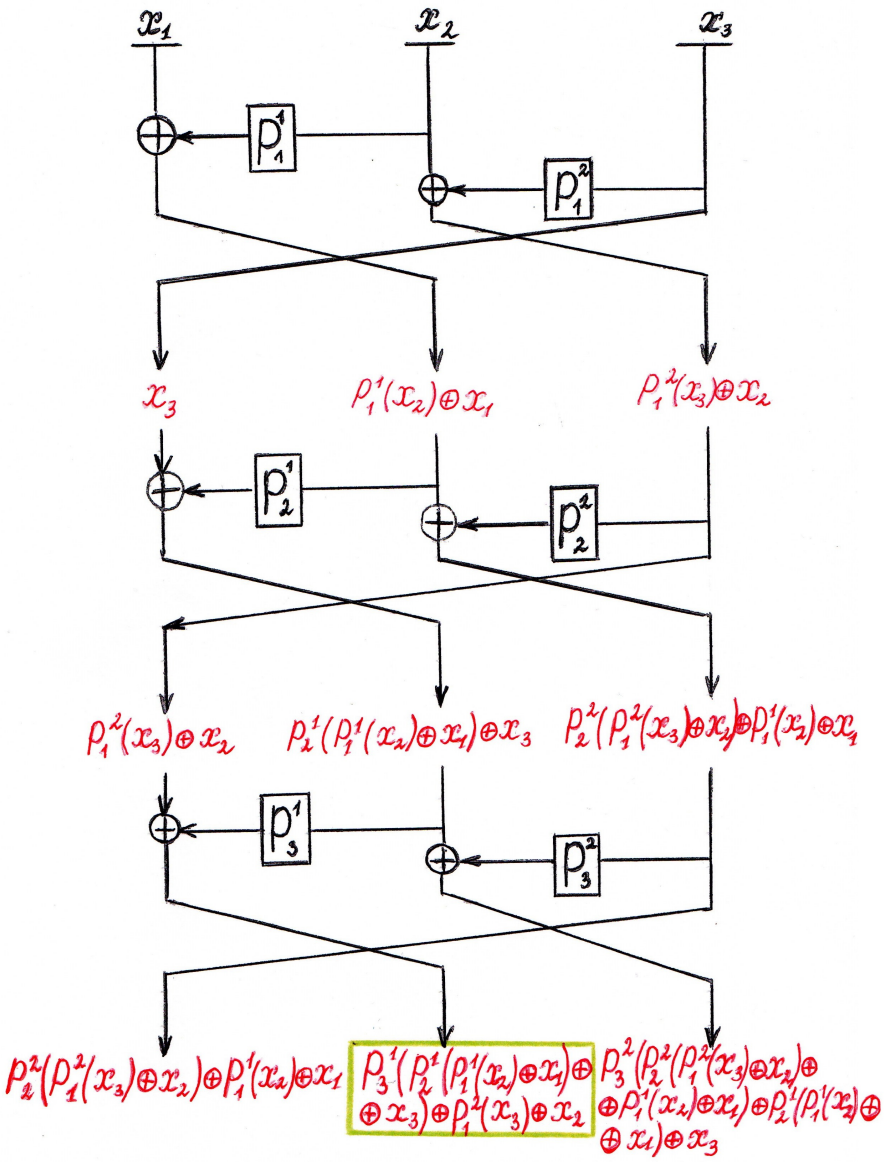


Рисунок 3.1 – Узагальнена схема Фейстеля типу III, де відкритий текст розбивається на три частини

Введемо функцію f, яка приймає вектор із множини $(2n+1)$-бітових векторів, а повертає вектор із множини n-бітових векторів. Функція f працює таким чином (α та β є деякими n-бітовими фіксованими векторами, x_3 — фіксоване):

$$f(b \parallel x_1 \parallel x_3) = \begin{cases} W(x_1 \parallel \alpha \parallel x_3) \oplus \beta, & \text{якщо } b = 0, \\ W(x_1 \parallel \beta \parallel x_3) \oplus \alpha, & \text{якщо } b = 1. \end{cases}$$

Лема 3.1. *Вихідний шифротекст на третьому раунді узагальненої схеми Фейстеля типу III, де відкритий текст розбивається на три частини (див. рис. 3.1), не є псевдовипадковою перестановкою у квантовій моделі обчислень.*

Доведення. Покажемо, що $f(b \parallel x_1 \parallel x_3) = f(b \parallel x_1^{'} \parallel x_3)$ тоді та тільки тоді, коли $b^{'} = b \oplus 1$ та $x_1^{'} = x_1 \oplus a$, де період $a = P_1^1(\alpha) \oplus P_1^1(\beta)$. Доведемо необхідність. Припустимо, що $b = b^{'} = 0$, тоді:

$$f(0 \parallel x_1 \parallel x_3) = W(x_1 \parallel \alpha \parallel x_3) \oplus \beta = P_3^1(P_2^1(P_1^1(\alpha) \oplus x_1) \oplus x_3) \oplus \\ \oplus P_1^2(x_3) \oplus \alpha \oplus \beta,$$
$$f(0 \parallel x_1^{'} \parallel x_3) = W(x_1^{'} \parallel \alpha \parallel x_3) \oplus \beta = P_3^1(P_2^1(P_1^1(\alpha) \oplus x_1^{'}) \oplus x_3) \oplus \\ \oplus P_1^2(x_3) \oplus \alpha \oplus \beta.$$

Оскільки $f(0 \parallel x_1 \parallel x_3) = f(0 \parallel x_1^{'} \parallel x_3)$ та P_2^1 — випадкова перестановка, маємо:

$$x_1 = x_1^{'} \text{ та } b \parallel x_1 \parallel x_3 = b^{'} \parallel x_1^{'} \parallel x_3.$$

Якщо $b = 0$, $b^{'} = 1$, тоді

$$f(1 \parallel x_1^{'} \parallel x_3) = W(x_1^{'} \parallel \beta \parallel x_3) \oplus \alpha = \\ = P_3^1(P_2^1(P_1^1(\beta) \oplus x_1^{'}) \oplus x_3) \oplus P_1^2(x_3) \oplus \beta \oplus \alpha.$$

Оскільки $f(0 \parallel x_1 \parallel x_3) = f(1 \parallel x_1^{'} \parallel x_3)$ та P_2^1 — випадкова перестановка, маємо:

$$x_1^{'} = x_1 \oplus a, \text{ де період } a = P_1^1(\alpha) \oplus P_1^1(\beta).$$

Аналогічне доведення для випадків, коли $b = 1$, $b^{'} = 1$ та $b = 1, b^{'} = 0$.

Доведемо достатність. Нехай $b = 1, b^{'} = 0$, тоді:

$$f(1 \parallel x_1^{'} \parallel x_3) = W(x_1^{'} \parallel \beta \parallel x_3) \oplus \alpha = P_3^1(P_2^1(P_1^1(\beta) \oplus x_1^{'}) \oplus x_3) \oplus P_1^2(x_3) \oplus$$
$$\oplus \beta \oplus \alpha = P_3^1(P_2^1(P_1^1(\beta) \oplus x_1 \oplus P_1^1(\alpha) \oplus P_1^1(\beta)) \oplus x_3) \oplus P_1^2(x_3) \oplus \beta \oplus \alpha =$$
$$= f(0 \parallel x_1 \parallel x_3).$$

Аналогічне можна довести для випадку, коли $b = 0$, $b^{'} = 1$. □

Тепер розглянемо схему на рисунку 3.2 і прослідкуємо виходи з кожного її раунду. Нехай вхідним повідомленням до даної схеми є $4n$-бітовий вектор

$$x = x_1 \parallel x_2 \parallel x_3 \parallel x_4, \text{ де } x_1,x_2,x_3,x_4 \in \{0,1\}^n.$$

Функції перестановки так само як у попередньому випадку позначимо P_i^j, де i — номер раунду, j — номер функції перестановки на раунді. З рисунку 3.2 бачимо, що вихідний другий блок на четвертому раунді має вигляд:

$$W(x) = W(x_1 \parallel x_2 \parallel x_3 \parallel x_4) = P_4^1(P_3^1(P_2^1(P_1^1(x_2) \oplus x_1) \oplus x_4) \oplus$$
$$\oplus P_1^3(x_4) \oplus x_3) \oplus P_2^3(P_1^3(x_4) \oplus x_3) \oplus P_1^2(x_3) \oplus x_2.$$

Перевизначимо функцію f. Тепер вона приймає вектор із множини $(3n+1)$-бітових векторів, а повертає вектор із множини n-бітових векторів. Функція f працює таким чином (α та β є деякими n-бітовими фіксованими векторами, x_3,x_4 — фіксовані):

$$f(b \parallel x_1 \parallel x_3 \parallel x_4) = \begin{cases} W(x_1 \parallel \alpha \parallel x_3 \parallel x_4) \oplus \beta, & \text{якщо } b = 0, \\ W(x_1 \parallel \beta \parallel x_3 \parallel x_4) \oplus \alpha, & \text{якщо } b = 1. \end{cases}$$

Лема 3.2. *Вихідний шифротекст на четвертому раунді узагальненої схеми Фейстеля типу III, де відкритий текст розбивається на чотири частини (див. рис. 3.2), не є псевдовипадковою перестановкою у квантовій моделі обчислень.*

Доведення. Покажемо, що $f(b \parallel x_1 \parallel x_3 \parallel x_4) = f(b \parallel x_1^{'} \parallel x_3 \parallel x_4)$ тоді та тільки тоді, коли $b^{'} = b \oplus 1$ та $x_1^{'} = x_1 \oplus a$, де період $a = P_1^1(\alpha) \oplus P_1^1(\beta)$.

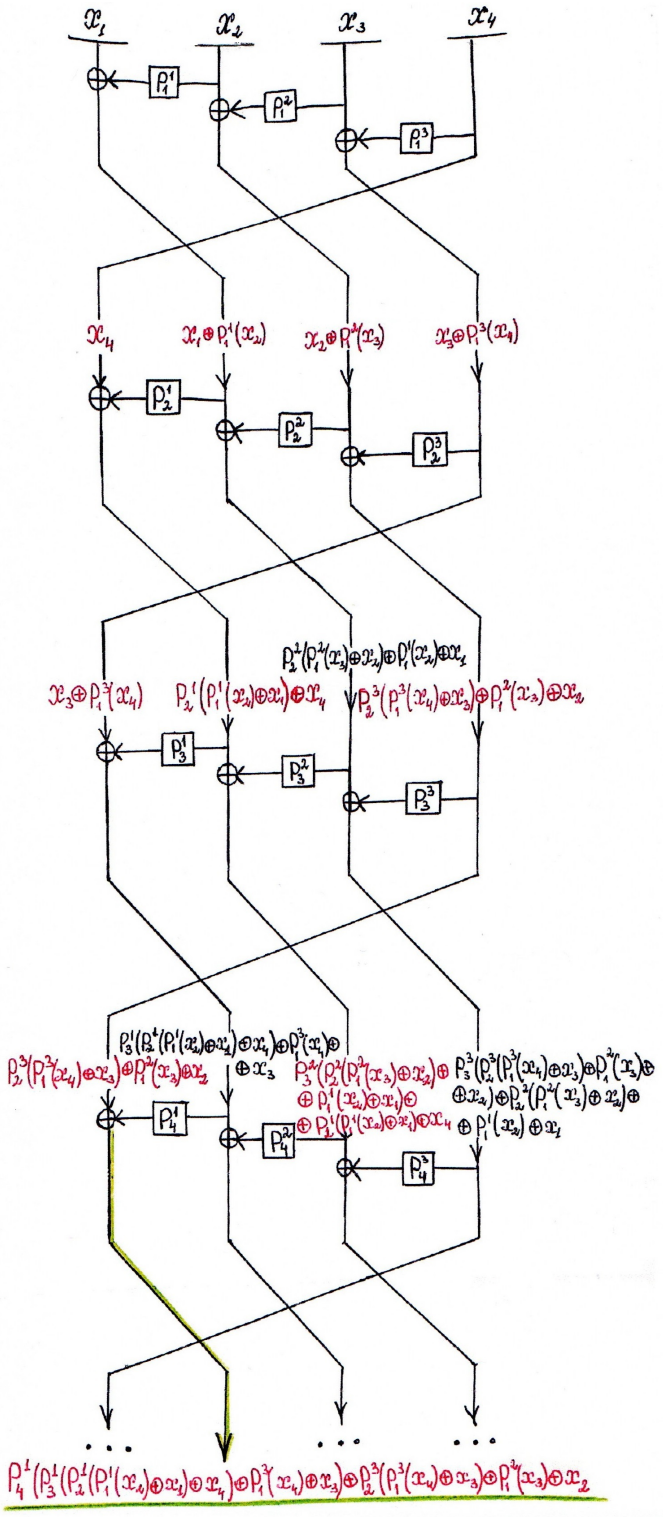


Рисунок 3.2 – Узагальнена схема Фейстеля типу III, де відкритий текст розбивається на чотири частини

Для зручності позначимо $K = P_1^3(x_4) \oplus x_3$, адже x_3 та x_4 є фіксованими. Доведемо необхідність. Припустимо, що $b = b' = 0$, тоді:

$$f(0 \parallel x_1 \parallel x_3 \parallel x_4) = W(x_1 \parallel \alpha \parallel x_3 \parallel x_4) \oplus \beta = P_4^1(P_3^1(P_2^1(P_1^1(\alpha) \oplus x_1) \oplus x_4) \oplus \oplus K) \oplus P_2^3(K) \oplus P_1^2(x_3) \oplus \alpha \oplus \beta,$$

$$f(0 \parallel x_1^{'} \parallel x_3 \parallel x_4) = W(x_1^{'} \parallel \alpha \parallel x_3 \parallel x_4) \oplus \beta = P_4^1(P_3^1(P_2^1(P_1^1(\alpha) \oplus x_1^{'}) \oplus x_4) \oplus \oplus K) \oplus P_2^3(K) \oplus P_1^2(x_3) \oplus \alpha \oplus \beta.$$

Оскільки $f(0 \parallel x_1 \parallel x_3 \parallel x_4) = f(0 \parallel x_1^{'} \parallel x_3 \parallel x_4)$ та P_2^1 — випадкова перестановка, маємо:

$$x_1 = x_1^{'} \text{ та } b \parallel x_1 \parallel x_3 \parallel x_4 = b^{'} \parallel x_1^{'} \parallel x_3 \parallel x_4.$$

Якщо $b = 0$, $b^{'} = 1$, тоді:

$$f(1 \parallel x_1^{'} \parallel x_3 \parallel x_4) = W(x_1^{'} \parallel \beta \parallel x_3 \parallel x_4) \oplus \alpha = P_4^1(P_3^1(P_2^1(P_1^1(\beta) \oplus x_1^{'}) \oplus x_4) \oplus \oplus K) \oplus P_2^3(K) \oplus P_1^2(x_3) \oplus \beta \oplus \alpha.$$

Оскільки $f(0 \parallel x_1) = f(1 \parallel x_1^{'})$ та P_2^1 — випадкова перестановка, маємо:

$$x_1^{'} = x_1 \oplus a, \text{ де період } a = P_1^1(\alpha) \oplus P_1^1(\beta).$$

Аналогічне доведення для випадків, коли $b = 1$, $b^{'} = 1$ та $b = 1$, $b^{'} = 0$.

Доведемо достатність. Нехай $b = 1, b^{'} = 0$, тоді:

$$f(0 \parallel x_1^{'} \parallel x_3 \parallel x_4) = W(x_1^{'} \parallel \alpha \parallel x_3 \parallel x_4) \oplus \beta = P_4^1(P_3^1(P_2^1(P_1^1(\alpha) \oplus x_1^{'}) \oplus x_4) \oplus \oplus K) \oplus P_2^3(K) \oplus P_1^2(x_3) \oplus \alpha \oplus \beta = P_4^1(P_3^1(P_2^1(P_1^1(\alpha) \oplus x_1 \oplus P_1^1(\alpha) \oplus P_1^1(\beta)) \oplus x_4) \oplus \oplus K) \oplus P_2^3(K) \oplus P_1^2(x_3) \oplus \alpha \oplus \beta = f(1 \parallel x_1 \parallel x_3 \parallel x_4).$$

Аналогічне можна довести для випадку, коли $b = 0, b^{'} = 1$. □

Наслідок 3.4. *З лем 3.1 та 3.2 стає зрозуміло, що d-раундова узагальнена схема Фейстеля типу III, де відкритий текст розбивається на d частин, не є стійкою до атак розрізнення випадкового тексту від шифротексту.*

Варто зазначити, що спроби побудувати аналогічні атаки для схеми Лая-Мессі, L-схеми, R-схеми, комбінацій L-, R- та B-схем, класичної чотирираундової схеми Фейстеля, а також збільшити кількість раундів узагальнених схем Фейстеля не дали результатів. Для таких схем потрібна інша ідея побудови функції f, адже інакше для знаходження її періоду треба знати обернені функції до перестановок шифру.

3.3 Застосування ротаційного криптоаналізу до шифру «Кипарис»

Для проведення ротаційного криптоаналізу шифру «Кипарис» вважатимемо, що його схема розгортання ключів є ідеальною, тобто кожний раундовий ключ є випадковим. Таке припущення є стандартним. Окрім цього, схема розгортання ключів обраного шифру схожа на схему розгортання ключів шифру «Калина», на яку поки що ніхто не зробив успішної атаки. Таким чином, додавання з ключем вводить у стан (тобто слово) певне перемішування (англ. confusion), тому раунди «Кипарис» приймаємо за незалежні.

Підрахуємо ймовірність ротації на першому раунді обраного шифру. Спочатку варто зазначити, що розглядати окремо різні версії «Кипарис» немає сенсу, адже на практиці показано, що для 32- та 64-бітових слів двійкові логарифми ймовірності ротації при параметрах $r = 1$ та $r = 2$ збігаються з точністю до однієї тисячної. Впевнитись у цьому можна у додатку Б. Отже, будемо користуватись округленими значеннями до десятих з таблиці 2.1 як для 32-бітових, так і для 64-бітових слів, що відповідають версіям «Кипарис-256» та «Кипарис-512» відповідно. «Кипарис» є класичною схемою Фейстеля, тому, якщо відкритим текстом у схему шифрування є

$$x = L_0 \parallel R_0 \in \{0,1\}^n,$$

де $L_0, R_0 \in \{0,1\}^{\frac{n}{2}}$ — ліва та права частини відкритого тексту, то функцію h^2 на першому раунді застосуємо тільки до однієї частини, а саме до L_0 (див. рис. 1.3). На i-тому ж раунді — до

$$L_i = R_{i-1} \oplus F(L_{i-1}, RK^{(i-1)}), \text{ де } R_i = L_{i-1}.$$

Функція h^2 містить два «ланцюги» додавань за модулем $2^{\frac{n}{2}}$. На рисунку 1.3 вони позначені рожевим та синім кольорами. Кожний ланцюг має чотири такі операції. Скориставшись теоремою про послідовні додавання

за модулем (теоремою 2.5), отримаємо, що ймовірність ротації для одного ланцюга дорівнює $2^{-9.3}$, якщо параметр ротації $r = 1$ (далі розглядаємо саме його). Оскільки ланцюги ніяк не пов'язані між собою, то можна вважати їх незалежними. Також, враховуючи те, що ліва частина відкритого тексту L_0 є лівою частиною вихідного шифротексту з першого раунду (ми знаємо її з ймовірністю 1), то ймовірність ротації для однораундового шифру складає

$$2^{-9.3 \cdot 2} \cdot 1 = 2^{-18.6}.$$

На другому раунді маємо текст $L_1 \parallel R_1$, де R_1 фактично є лівою частиною відкритого тексту L_0, а L_1 — правим виходом із першого раунду шифру, ймовірність якого порахована вище. Враховуючи те, що ймовірність входу L_1 до функції h^2 складає $2^{-18.6}$, то за теоремою про незалежні додавання за модулем (теоремою 2.4) та теоремою про послідовні додавання за модулем (теоремою 2.5) маємо таку ймовірність ротації виходу з цієї функції:

$$2^{-18.6} \cdot 2^{-18.6} = (2^{-18.6})^2 = 2^{-37.2}.$$

Тоді для двох раундів шифру «Кипарис» ймовірність ротації складає

$$(2^{-18.6})^2 \cdot 2^{-18.6} = (2^{-18.6})^3 = 2^{-55.8},$$

де другим множником є ймовірність лівої частини L_1. Відповідно, на третьому раунді для лівої частини маємо $(2^{-18.6})^2 = 2^{-37.2}$, коли точну ймовірність правої обрахувати складніше, адже вона є ймовірністю результату застосування операції виключної диз'юнкції до значень $h^2(L_2)$ та R_2, де ймовірність значення $h^2(L_2)$ складає $2^{-55.8}$, а R_2 — $2^{-18.6}$. Таким чином, маємо справу з нерівномірно розподіленими $\frac{n}{2}$-бітовими векторами, тобто з нерівномірно розподіленими значеннями самих бітів у векторах. Більш того, розподіл векторів $h^2(L_2)$ та R_2 може бути різним, тому шукати ймовірність результату як добуток їх ймовірностей не можна. Для кращого розуміння розглянемо приклад.

Приклад 3.1. Нехай є дві множини трьохбітових двійкових векторів виду (x_2^i,x_1^i,x_0^i), де i є номером множини (I або II). Вважаємо, що у першій множині ймовірність значення 0 на молодшій позиції дорівнює $\frac{1}{3}$, на середній — $\frac{4}{5}$, а на старшій — $\frac{1}{6}$, тобто:

$$Pr[x_2^I = 0] = \tfrac{1}{6},\ Pr[x_1^I = 0] = \tfrac{4}{5},\ Pr[x_0^I = 0] = \tfrac{1}{3}.$$

На другій множині розподіл задаємо таким:

$$Pr[x_2^{II} = 0] = \tfrac{7}{9},\ Pr[x_1^{II} = 0] = \tfrac{1}{4},\ Pr[x_0^{II} = 0] = \tfrac{2}{3}.$$

Тоді ймовірність вектора (1,0,0) із припущення про незалежність бітів на першій множині складає

$$\tfrac{5}{6} \cdot \tfrac{4}{5} \cdot \tfrac{1}{3} = \tfrac{2}{9},$$

коли на другій множині ймовірність (1,0,0) дорівнює

$$\tfrac{2}{9} \cdot \tfrac{1}{4} \cdot \tfrac{2}{3} = \tfrac{1}{27}.$$

Тепер підрахуємо ймовірність, що деякий вектор z, який є результатом застосування операції виключної диз'юнкції до векторів x^I, x^{II} із першої та другої множин відповідно, рівний вектору (0,0,0).

$$\begin{gathered}
Pr[z = (0,0,0), z = x^I \oplus x^{II}] = Pr[x^I = (0,0,0)] \cdot Pr[x^{II} = (0,0,0)] + \\
+ Pr[x^I = (0,0,1)] \cdot Pr[x^{II} = (0,0,1)] + Pr[x^I = (0,1,0)] \cdot Pr[x^{II} = (0,1,0)] + \\
+ Pr[x^I = (0,1,1)] \cdot Pr[x^{II} = (0,1,1)] + Pr[x^I = (1,0,0)] \cdot Pr[x^{II} = (1,0,0)] + \\
+ Pr[x^I = (1,0,1)] \cdot Pr[x^{II} = (1,0,1)] + Pr[x^I = (1,1,0)] \cdot Pr[x^{II} = (1,1,0)] + \\
+ Pr[x^I = (1,1,1)] \cdot Pr[x^{II} = (1,1,1)] = ... = \tfrac{119}{2430} \approx 0.049,
\end{gathered}$$

коли один доданок $Pr[x^I = (1,0,0)] \cdot Pr[x^{II} = (1,0,0)]$ дав би $\frac{2}{243} \approx 0.0082$. Тобто доданками нехтувати не можна. Більш того, зі збільшенням довжини векторів матимемо більше доданків, що призводить до ще більшої різниці між коректним значенням ймовірності та деяким його доданком.

Варто зазначити, що розподіл вихідних векторів (лівої й правої частин) з раундів шифрування нам невідомий, тому прямі обрахунки як у прикладі 3.1 зробити просто неможливо. Однак ймовірність виходу правої частини на третьому раунді можна обчислити з припущенням, що

ймовірність R_2 складає 1, адже тоді ймовірність вихідного значення правої частини з третього раунду є просто ймовірністю $h^2(L_2)$. Таким чином, у криптоаналітика з'явилось більше шансів для атаки ніж при прямих обрахунках. Тоді на третьому раунді ймовірність ротації дорівнює

$$(2^{-18.6})^2 \cdot (2^{-18.6})^3 = (2^{-18.6})^5,$$

де $(2^{-18.6})^2$ є ймовірністю лівої частини L_2, а $(2^{-18.6})^3$ — правої.

Якщо продовжити розписувати ймовірності ротації виходів з кожного раунду шифру «Кипарис», то можна побачити просту формулу обрахунку ймовірності ротації для i раундів:

$$(2^{-18.6})^{2i-1}.$$

На рисунку 3.3 наведена візуалізація чотирнадцяти раундів шифру, де синім кольором позначені ймовірності ротації лівої й правої частин тексту, які поступають до того чи іншого раунду схеми, а зеленим — ймовірність, яку набуває ліва частина тексту при проходженні функції h^2.

Тоді для «Кипарис-256», за специфікацією якого маємо 10 раундів, отримаємо, що ймовірність ротації складає

$$(2^{-18.6})^{19} = 2^{-353.4},$$

коли ймовірність випадкової перестановки на 256-бітових векторах 2^{-256}. Тобто «Кипарис-256» є стійким до ротаційного криптоаналізу. Окрім цього, дана версія стійка і до RX-криптоаналізу, адже останній окрім зсуву використовує деякі константи, що введені у стан. Якщо простий циклічний зсув на один біт відстежити не можна, то, застосовуючи операцію виключної диз'юнкції до результату зсуву, також нічого відстежити не зможемо.

Для «Кипарис-512» маємо 14 раундів, тому ймовірність є наступною:

$$(2^{-18.6})^{27} = 2^{-502.2}.$$

Цього разу отримана ймовірність є більшою за ймовірність випадкової перестановки, яка тепер складає 2^{-512}, однак близькою до неї, тому «Кипарис-512» є стійким до R- та RX-криптоаналізів.

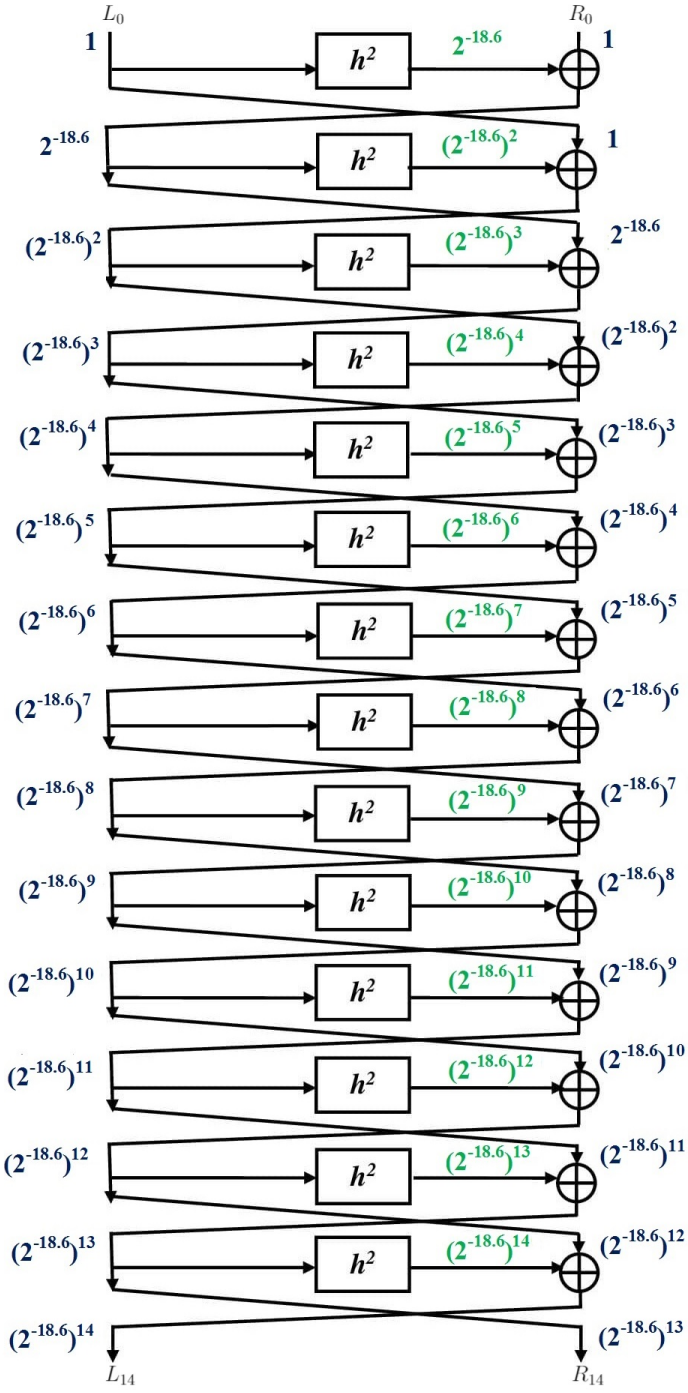


Рисунок 3.3 – Вихідні ймовірності з кожного з чотирнадцяти раундів шифру «Кипарис»

Якщо у схему шифрування додати ще один раунд, то матимемо ймовірність рівну $2^{-539.4}$, а при додаванні ключа до правої частини на останньому раунді, знову використовуючи припущення про ідеальний ключовий розподіл, отримаємо ймовірність

$$(2^{-18.6})^{14} \cdot 2^{-256} = 2^{-516.4},$$

де перший множник відповідає ймовірності ротації лівої частини, а другий — випадкової перестановки на 256-бітових векторах.

Якщо розглянути параметр ротації $r = 2$, то за аналогією до наведених вище обчислень, користуючись таблицею 2.1, для «Кипарис-512» отримаємо:

$$(2^{-22.2})^{27} = 2^{-599.4},$$

тобто «Кипарис-512» має стійкість до ротації на 2 біти.

З підрозділу 2.2 зрозуміло, що при послідовних додаваннях за модулем маємо меншу ймовірність ніж при обрахунках, коли між ними наявні інші операції. Так застосування операцій виключної диз'юнкції з ключем у раундовій функції та виключної диз'юнкції до лівої частини минулого раунду і правої поточного може й не реалізовувати випадкову перестановку, тому якщо розглянемо шифр «Кипарис», у якому використовується тільки функція h, то для версії «Кипарис-256» отримаємо два ланцюги по $8 \cdot 10$ модульних додавань кожний для лівої частини та два ланцюги по $8 \cdot 9$ модульних додавань кожний для правої частини. Тобто, використовуючи таблицю Б.1, маємо, що ймовірність «Кипарис-256» при параметрі ротації $r = 1$ складає

$$(2^{-475.8})^2 \cdot (2^{-417.8})^2 = 2^{-1787.2}.$$

Для «Кипарис-512» маємо два ланцюги по $8 \cdot 14$ модульних додавань кожний для лівої частини та два по $8 \cdot 13$ модульних додавань для правої частини. Таким чином, використовуючи таблицю Б.1, отримаємо

$$(2^{-718.6})^2 \cdot (2^{-656.5})^2 = 2^{-2750.2}.$$

Такі підрахунки кажуть про те, що прибравши операцію виключної диз'юнкції з ключем на кожному раунді, а також виключну диз'юнкцію правої та лівої частин у схемі «Кипарис», ротаційні атаки унеможливлюються, але можуть з'являтись інші. Прибирати з даного шифру нічого не треба, адже у цьому немає потреби, а ймовірності ротації складають $2^{-353.4}$ та $2^{-502.2}$ для 256-бітової та 512-бітової версій відповідно. Для останньої можна додати ключ у кінці.

Висновки до розділу 3

Сьогодні існує величезна кількість методів та підходів до аналізу тих чи інших шифрів. У роботі предметом дослідження обрано стійкість малоресурсного симетричного блокового шифру «Кипарис» до квантового та ротаційного криптоаналізів, яка вперше досліджувалась у даному розділі. У квантовій моделі обчислень зроблена поліноміальна атака розрізнення випадкового тексту від шифротексту на трираундову схему шифрування «Кипарис», адже задача пошуку періоду такої схеми, де два різні вхідні тексти мають однаковий шифротекст, є задачею Саймона, яка має ефективний розв'язок у квантовій моделі. Більш того, застосувавши комбінацію алгоритмів Саймона та Гровера, зроблена субекспоненційна атака на п'ять раундів даного шифру. Оцінки стійкості більшої кількості раундів шифру до квантового криптоаналізу наразі є експоненційними.

Також вперше доведена вразливість узагальненої схеми Фейстеля типу III у квантовій моделі обчислень, для успішної атаки розрізнення випадкового тексту від шифротексту d раундів якої необхідно та достатньо, щоб відкритий текст на вході поділявся на d частин.

Окрім цього, вперше проведено ротаційний криптоаналіз шифру «Кипарис» за припущенням про ідеальність схеми розгортання ключів. За результатами отримано, що ймовірність проведення ротаційного криптоаналізу для версії «Кипарис-256» складає $2^{-353.4}$, а для «Кипарис-512» — $2^{-502.2}$, якщо параметр ротації встановити рівним 1. Тобто 256-бітова версія є стійкою як до R-криптоаналізу, так і до RX-криптоаналізу. Для 512-бітової версії, враховуючи зроблене припущення, ймовірність ротації можна зменшити, якщо на виході схеми шифрування застосувати операцію виключної диз'юнкції з ключем до правої частини або додати до схеми ще один раунд.

ВИСНОВКИ

Наявність великої кількості малоресурсних пристроїв сьогодні, а також стрімкий розвиток Інтернету речей стали поштовхом для появи «легкої криптографії», розробники якої повинні знайти компроміс між безпекою, витратами та ефективністю реалізації криптопримітивів. Для забезпечення швидкості обчислень зазвичай використовують операції додавання за модулем, виключної диз'юнкції та циклічного зсуву, а відповідні шифри називають ARX-шифрами. У першому розділі зроблено огляд наявних методів та підходів до їх криптоаналізу. Розглянуто основи диференціального, лінійного, ротаційного та квантового криптоаналізів, а також викладені ідеї слайд-атак, криптоаналізу на пов'язаних ключах, деяких комбінованих методів криптоаналізу та їх модифікацій. Окрім цього, розглянуті наявні ARX-шифри, зокрема малоресурсний симетричний блоковий шифр «Кипарис», стійкість якого є предметом дослідження роботи. Детальний аналіз наявних робіт з ротаційного криптоаналізу та опис квантових атак на фейстель-подібні схеми представлено у другому розділі.

Скориставшись результатами розглянутих робіт з квантового та ротаційного криптоаналізів, у третьому розділі вперше отримані такі результати:

1) побудована поліноміальна атака розрізнення випадкового тексту від шифротексту у квантовій моделі обчислень на три раунди схеми шифрування «Кипарис», використовуючи зведення до задачі Саймона;

2) побудована субекспоненційна атака відновлення ключів на четвертому та п'ятому раундах шифру «Кипарис», застосовуючи комбінацію алгоритмів Саймона та Гровера. Оцінки ймовірностей проведення аналогічних атак на всі раунди шифру на цей момент є експоненційними, що гарантує стійкість обраного шифру у постквантовий період;

3) доведена вразливість узагальненої схеми Фейстеля типу III у квантовій моделі обчислень, де кількість раундів збігається з кількістю частин, на які розбивається відкритий текст на вході у схему шифрування, використовуючи зведення до задачі Саймона;

4) доведена стійкість схеми шифрування «Кипарис-256» до ротаційного криптоаналізу з параметром ротації встановленим рівним одиниці, ймовірність успішного проведення якого складає $2^{-353.4}$, коли ймовірність випадкової перестановки — 2^{-256}, що, відповідно, передбачає стійкість як до ротаційного криптоаналізу з більшим параметром ротації, так і до RX-криптоаналізу. Оцінка ймовірності ротації для «Кипарис-512» становить $2^{-502.2}$, коли для випадкової перестановки — 2^{-512}. Для більшої гарантії неможливості проведення ротаційного криптоаналізу «Кипарис-512» рекомендовано після виходу з останнього раунду шифрування виконати операцію виключної диз'юнкції правої частини шифротексту з ключем, адже тоді ймовірність ротації складатиме $2^{-516.4}$. Також можна додати ще один раунд шифрування. Окрім цього, підрахована ймовірність ротації на 2 біти для «Кипарис-512», яка дорівнює $2^{-599.4}$.

У подальших роботах треба досліджувати стійкість шифру «Кипарис», використовуючи інші методи криптоаналізу, перевірити стійкість до квантового диференціального та лінійного криптоаналізів, а також до класичного лінійного криптоаналізу. Окрім цього, треба досліджувати та розробляти методи аналізу ARX-шифрів, поширювати наявні атаки на них.

ПЕРЕЛІК ПОСИЛАНЬ

1. Deepti Sehrawat and Nasib Singh Gill. *Lightweight Block Ciphers for IoT based applications: A Review* [Електронний ресурс] – 2018. – Режим доступу: https://www.ripublication.com/ijaer18/ijaerv13n5_26.pdf

2. Dimiter V. Dimitrov. *Medical Internet of Things and Big Data in Healthcare* [Електронний ресурс] – 2016. – Режим доступу: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4981575/

3. Tuan Nguyen Gia, Mai Ali, Imed Ben Dhaou, Amir M.Rahmani, Tomi Westerlund, Pasi Liljeberg, Hannu Tenhunen. *IoT-based continuous glucose monitoring system: A feasibility study* [Електронний ресурс] – 2017. – Режим доступу: https://www.sciencedirect.com/science/article/pii/S1877050917310281

4. Axel Poschmann. *Lightweight Cryptography - Cryptographic Engineering for a Pervasive World* [Електронний ресурс] – 2009. – Режим доступу: https://eprint.iacr.org/2009/516.pdf

5. Shimizu A., Miyaguchi S. *Fast Data Encipherment Algorithm FEAL* [Електронний ресурс] – 1988. – Режим доступу: https://link.springer.com/chapter/10.1007/3-540-39118-5_24.

6. Ralf-Philipp Weinmann. *ARX - Crypto made from modular additions, XORs and word rotations* [Електронний ресурс] – 24 February, 2009. – Режим доступу: https://fse2009rump.cr.yp.to/b8849fea34b9c80aad42a29aa1abe20f.pdf.

7. Jean-Philippe Aumasson, Samuel Neves, Zooko Wilcox-O'Hearn, and Christian Winnerlein. *BLAKE2: Simpler, Smaller, Fast as MD5* [Електронний ресурс] – 2003. – Режим доступу: https://eprint.iacr.org/2013/322.pdf

8. N. Ferguson, S. Lucks, B. Schneier, D. Whiting, M. Bellare, T. Kohno, J. Callas, and J. Walker. *The Skein Hash Function Family* [Електронний ресурс] – 2010. – Режим доступу: https://www.schneier.com/wp-content/uploads/2016/02/skein.pdf

9. R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. *The SIMON and SPECK lightweight block ciphers* [Електронний ресурс] – 2015. – Режим доступу: https://eprint.iacr.org/2013/404.pdf

10. Daniel Dinu, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, Johann Großschädl, Alex Biryukov. *Design Strategies for ARX with Provable Bounds: Sparx and LAX* [Електронний ресурс] – 2016. – Режим доступу: https://eprint.iacr.org/2016/984.pdf

11. Deukjo Hong, Jung-Keun Lee, Dong-Chan Kim, Daesung Kwon, Kwon Ho Ryu, and Dong-Geon Lee. *LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors* [Електронний ресурс] – 2014. – Режим доступу: https://link.springer.com/chapter/10.1007/978-3-319-05149-9_1#:~:text=We%20propose%20a%20new%20block,to%20have%20tiny%20code%20size.

12. Daniel J. Bernstein. *The Salsa20 family of stream ciphers* [Електронний ресурс] – 2008. – Режим доступу: https://link.springer.com/chapter/10.1007/978-3-540-68351-3_8

13. Daniel J. Bernstein. *ChaCha, a variant of Salsa20* [Електронний ресурс] – 2008. – Режим доступу: https://cr.yp.to/chacha/chacha-20080120.pdf

14. Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. *Chaskey: An Efficient MAC Algorithm for 32-bit Microcontrollers* [Електронний ресурс] – 2014. – Режим доступу: https://eprint.iacr.org/2014/386.pdf

15. Марія Родінко, Роман Олійников. *Постквантовий малоресурсний симетричний блоковий шифр «Кипарис»* [Електронний ресурс] – 2017. – Режим доступу: http://www.irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?I21DBN=LINK&P21DBN=UJRN&Z21ID=&S21REF=10&S21CNR=20&S21STN=1&S21FMT=ASP_meta&C21COM=S&2_S21P03=FILA=&2_S21STR=rvmnts_2017_189_13

16. Марія Родінко, Роман Олійников. *Дослідження продуктивності малоресурсного блокового шифру «Кипарис» на різних платформах*

[Електронний ресурс] – 2020. – Режим доступу: http://rt.nure.ua/article/view/210062/210121

17. Gangqiang Yang, Bo Zhu, Valentin Suder, Mark D. Aagaard, and Guang Gong. *The Simeck Family of Lightweight Block Ciphers* [Електронний ресурс] – 2015. – Режим доступу: https://eprint.iacr.org/2015/612.pdf

18. Jean-Philippe Aumasson, Philipp Jovanovic, and Samuel Neves. *NORX: Parallel and Scalable AEAD* [Електронний ресурс] – 2014. – Режим доступу: https://link.springer.com/chapter/10.1007/978-3-319-11212-1_2

19. Magnus Daum. *Cryptanalysis of Hash Functions of the MD4-Family* [Електронний ресурс] – 2005. – Режим доступу: https://www.cits.ruhr-uni-bochum.de/imperia/md/content/magnus/dissmd4.pdf

20. Dmitry Khovratovich and Ivica Nikoli´c. *Rotational Cryptanalysis of ARX* [Електронний ресурс] – 2010. – Режим доступу: https://link.springer.com/content/pdf/10.1007%2F978-3-642-13858-4_19.pdf

21. Souradyuti Paul and Bart Preneel. *Solving Systems of Differential Equations of Addition and Cryptanalysis of the Helix Cipher* [Електронний ресурс] – 2005. – Режим доступу: https://eprint.iacr.org/2004/294.pdf, 2005.

22. Bruce Schneier. *Applied Cryptography: Protocols, Algorithms and Source Code in C* - 2012.

23. Eli Biham and Adi Shamir. *Differential Cryptanalysis of DES-like Cryptosystems* [Електронний ресурс] – 1990. – Режим доступу: https://link.springer.com/content/pdf/10.1007/BF00630563.pdf

24. Eli Biham and Adi Shamir. *Differential Cryptanalysis of Feal and N-Hash* [Електронний ресурс] – 1991. – Режим доступу: https://link.springer.com/content/pdf/10.1007%2F3-540-46416-6_1.pdf

25. Xuejia Lai, James L. Massey. *Markov Ciphers and Differential Cryptanalysis* [Електронний ресурс] – 1991. – Режим доступу: https://link.springer.com/chapter/10.1007/3-540-46416-6_2

26. Mitsuru Matsui Atsuhiro Yamagishi. *A New Method for Known Plaintext Attack of FEAL Cipher* [Електронний ресурс] – 1992. – Режим доступу: https://link.springer.com/content/pdf/10.1007%2F3-540-47555-9_7.

pdf

27. Mitsuru Matsui. *Linear Cryptanalysis Method for DES Cipher* [Електронний ресурс] – 1993. – Режим доступу: https://link.springer.com/content/pdf/10.1007%2F3-540-48285-7_33.pdf

28. Dmitry Khovratovich, Ivica Nikoli´c, Josef Pieprzyk, Przemys law Soko lowski, Ron Steinfeld. *Rotational Cryptanalysis of ARX Revisited* [Електронний ресурс] – 2015. – Режим доступу: https://eprint.iacr.org/2015/095.pdf

29. Dmitry Khovratovich, Ivica Nikoli´c, Christian Rechberger. *Rotational Rebound Attacks on Reduced Skein* [Електронний ресурс] – 2010. – Режим доступу: https://eprint.iacr.org/2010/538.pdf

30. Standaert, F.X., Piret, G., Gershenfeld, N., Quisquater, J.J. *SEA: A Scalable Encryption Algorithm for Small Embedded Applications* [Електронний ресурс] – 2006. – Режим доступу: https://link.springer.com/chapter/10.1007/11733447_16

31. Tomer Ashur, Yunwen Liu. *Rotational Cryptanalysis in the Presence of Constants* [Електронний ресурс] – 2016. – Режим доступу: https://eprint.iacr.org/2016/826.pdf

32. Ernst Schulte-Geers. *On CCZ-equivalence of Addition mod* 2^n *Ernst Schulte-Geers* [Електронний ресурс] – 2013. – Режим доступу: https://hal.inria.fr/inria-00607747/document

33. Yunwen Liu, Glenn De Witte, Adrián Ranea, and Tomer Ashur. *Rotational-XOR Cryptanalysis of Reduced-round SPECK* [Електронний ресурс] – 2017. – Режим доступу: https://eprint.iacr.org/2017/1036.pdf

34. Adri´an Ranea, Yunwen Liu, Tomer Ashur. *An Easy-To-Use Tool for Rotational-XOR Cryptanalysis of ARX Block Ciphers* [Електронний ресурс] – 2020. – Режим доступу: https://eprint.iacr.org/2020/727.pdf

35. Stefano Barbero, Emanuele Bellini, and Rusydi Makarim. *Rotational analysis of ChaCha permutation* [Електронний ресурс] – 2020. – Режим доступу: https://eprint.iacr.org/2020/1049.pdf

36. Liliya Kraleva, Tomer Ashur, Vincent Rijmen. *Rotational*

Cryptanalysis on MAC Algorithm Chaskey [Електронний ресурс] – 2020. – Режим доступу: https://eprint.iacr.org/2020/538.pdf

37. Bonwook Koo, Younghoon Jung, and Woo-Hwan Kim. *Rotational-XOR Rectangle Cryptanalysis on Round-Reduced Simon* [Електронний ресурс] – 2020. – Режим доступу: https://www.hindawi.com/journals/scn/2020/5968584/

38. Jinyu Lu, Yunwen Liu, Tomer Ashur, Bing Sun, and Chao Li. *Rotational-XOR Cryptanalysis of Simon-Like Block Ciphers* [Електронний ресурс] – 2020. – Режим доступу: https://eprint.iacr.org/2020/486.pdf

39. Jonathan P. Dowling and Gerard J. Milburn. *Quantum technology: the second quantum revolution* [Електронний ресурс] – 2003. – Режим доступу: https://arxiv.org/ftp/quant-ph/papers/0206/0206091.pdf

40. NIST. *Introduction: A New Quantum Revolution* [Електронний ресурс] – 2018. – Режим доступу: https://www.nist.gov/topics/physics/introduction-new-quantum-revolution

41. NIST. *The Second Quantum Revolution* [Електронний ресурс] – 2018. – Режим доступу: https://www.nist.gov/topics/physics/introduction-new-quantum-revolution/second-quantum-revolution#:~:text=The%20second%20quantum%20revolution%20is,an%20experimental%20NIST%20atomic%20clock

42. Kaggle. *VinBigData Chest X-ray Abnormalities Detection* [Електронний ресурс] – 2021. – Режим доступу: https://www.kaggle.com/c/vinbigdata-chest-xray-abnormalities-detection/overview/prizesk

43. Richard P. Feynman. *Simulating physics with computers* [Електронний ресурс] – 1982. – Режим доступу: https://vql.cs.msu.ru/Feynman.pdf

44. David Deutsch and Artur Ekert. *Machines, Logic and Quantum Physics* [Електронний ресурс] – 1999. – Режим доступу: https://arxiv.org/pdf/math/9911150.pdf

45. D-Wave. *D-Wave Announces General Availability of First Quantum Computer Built for Business* [Електронний ресурс] –

2020. – Режим доступу: https://www.dwavesys.com/press-releases/d-wave-announces-general-availability-first-quantum-computer-built-business

46. Peter Shor. *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer* [Електронний ресурс] – 1994. – Режим доступу: http://mmrc.amss.cas.cn/tlb/201702/W020170224608150589788.pdf

47. Ethan Bernstein, Umesh Vazirani. *Quantum Complexity Theory* [Електронний ресурс] – 1993. – Режим доступу: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.655.1186&rep=rep1&type=pdf

48. Daniel Simon. *On the Power of Quantum Computation* [Електронний ресурс] – 1994. – Режим доступу: https://ieeexplore.ieee.org/document/365701

49. Lov K. Grover. *A fast quantum mechanical algorithm for database search* [Електронний ресурс] – 1996. – Режим доступу: https://arxiv.org/pdf/quant-ph/9605043.pdf

50. Kuwakado Hidenori, Morii Masakatu. *Quantum Distinguisher Between the 3-Round Feistel Cipher and the Random Permutation* [Електронний ресурс] – 2010. – Режим доступу: https://ieeexplore.ieee.org/document/5513654

51. Xiaoyang Dong, Zheng Li, and Xiaoyun Wang. *Quantum cryptanalysis on some Generalized Feistel Schemes* [Електронний ресурс] – 2017. – Режим доступу: https://eprint.iacr.org/2017/1249.pdf

52. Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, María Naya-Plasencia. *Breaking Symmetric Cryptosystems using Quantum Period Finding* [Електронний ресурс] – 2016. – Режим доступу: https://arxiv.org/abs/1602.05973

53. Gregor Leander and Alexander May. *Grover Meets Simon – Quantumly Attacking the FX-construction* [Електронний ресурс] – 2017. – Режим доступу: https://eprint.iacr.org/2017/427.pdf

54. Xiaoyang Dong and Xiaoyun Wang. *Quantum key-recovery attack on Feistel structures* [Електронний ресурс] – 2017. – Режим доступу: https:

//eprint.iacr.org/2017/1199.pdf

55. Xiaoyang Dong, Bingyou Dong, Xiaoyun Wang. *Quantum Attacks on Some Feistel Block Ciphers* [Електронний ресурс] – 2018. – Режим доступу: https://eprint.iacr.org/2018/504.pdf

56. Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, María Naya-Plasencia. *Quantum Differential and Linear Cryptanalysis* [Електронний ресурс] – 2015. – Режим доступу: https://arxiv.org/abs/1510.05836

57. Huiqin Xie, Li Yang. *Quantum differential cryptanalysis to the block ciphers* [Електронний ресурс] – 2015. – Режим доступу: https://arxiv.org/abs/1511.08800

58. Huiqin Xie, Li Yang. *Using Bernstein-Vazirani algorithm to attack block ciphers* [Електронний ресурс] – 2018. – Режим доступу: https://arxiv.org/abs/1711.00853

59. Xavier Bonnetain, María Naya-Plasencia, and André Schrottenloher. *Quantum Security Analysis of AES* [Електронний ресурс] – 2019. – Режим доступу: https://eprint.iacr.org/2019/272.pdf

60. Марія Родінко. *Оцінка стійкості симетричного блокового шифру «Кипарис» до диференційного криптоаналізу* [Електронний ресурс] – 2018. – Режим доступу: https://nure.ua/wp-content/uploads/2018/Scientific_editions/rvmnts_2018_195_13.pdf

61. David Wagner. *The Boomerang Attack* [Електронний ресурс] – 1999. – Режим доступу: https://link.springer.com/content/pdf/10.1007%2F3-540-48519-8_12.pdf

62. John Kelsey, Tadayoshi Kohno, and Bruce Schneier. *Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent* [Електронний ресурс] – 2000. – Режим доступу: https://link.springer.com/chapter/10.1007/3-540-44706-7_6

63. Eli Biham, Orr Dunkelman, and Nathan Keller. *The Rectangle Attack – Rectangling the Serpent* [Електронний ресурс] – 2001. – Режим доступу: https://link.springer.com/chapter/10.1007/3-540-44987-6_21

64. Susan K. Langford and Martin E. Hellman. *Differential-Linear*

Cryptanalysis [Електронний ресурс] – 1994. – Режим доступу: https://link.springer.com/chapter/10.1007/3-540-48658-5_3

65. Christof Beierle, Gregor Leander, and Yosuke Todo. *Improved Differential-Linear Attacks with Applications to ARX Ciphers* [Електронний ресурс] – 2020. – Режим доступу: https://eprint.iacr.org/2020/775.pdf

66. Yunwen Liu, Zhongfeng Niu, Siwei Sun, Chao Li, and Lei Hu. *Rotational Cryptanalysis From a Differential-linear Perspective: Practical Distinguishers for Round-reduced FRIET, Xoodoo, and Alzette* [Електронний ресурс] – 2021. – Режим доступу: https://eprint.iacr.org/2021/189.pdf

67. Yunwen Liu, Siwei Sun, and Chao Li. *A Note on the Bias of Rotational Differential-Linear Distinguishers* [Електронний ресурс] – 2021. – Режим доступу: https://eprint.iacr.org/2021/379.pdf

68. Alex Biryukov and David Wagner. *Slide Attacks* [Електронний ресурс] – 1999. – Режим доступу: https://link.springer.com/content/pdf/10.1007/3-540-48519-8_18.pdf

69. Eli Biham. *New Types of Cryptanalytic Attacks Using Related Keys* [Електронний ресурс] – 1993. – Режим доступу: https://link.springer.com/content/pdf/10.1007%2F3-540-48285-7_34.pdf

70. Michael Luby and Charles Rackoff. *How to Construct Pseudorandom Permutations from Pseudorandom Functions* [Електронний ресурс] – 1988. – Режим доступу: https://inst.eecs.berkeley.edu/~cs276/fa20/notes/Luby_Rackoff_paper.pdf

71. Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. *Quantum Amplitude Amplification and Estimation* [Електронний ресурс] – 2 May, 2000. – Режим доступу: https://arxiv.org/pdf/quant-ph/0005055.pdf

72. Viet Tung Hoang and Phillip Rogaway. *On Generalized Feistel Networks* [Електронний ресурс] – 2010. – Режим доступу: https://eprint.iacr.org/2010/301.pdf

73. Kaisa Nyberg and Johan Wallén, J. *Improved linear distinguishers for SNOW 2.0* [Електронний ресурс] – 2006. – Режим доступу: https://link.springer.com/chapter/10.1007/11799313_10

74. Lars R. Knudsen, Vincent Rijmen, Ronald L. Rivest, and Matthew J. B. Robshaw. *On the design and security of RC2* [Електронний ресурс] – 1998. – Режим доступу: https://link.springer.com/chapter/10.1007/3-540-69710-1_14

75. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. *Notes on the design and analysis of Simon and Speck* [Електронний ресурс] – 2017. – Режим доступу: https://eprint.iacr.org/2017/560.pdf

76. Ling Song, Zhangjie Huang, and Qianqian Yang. *Automatic Differential Analysis of ARX Block Ciphers with Application to SPECK and LEA* [Електронний ресурс] – 2016. – Режим доступу: https://eprint.iacr.org/2016/209.pdf

77. Роман Елисеев, Мария Родинко, Роман Олейников. *Дифференциальный криптоанализ блочного ARX-шифра «Кипарис-256»* [Електронний ресурс] – 2018. – Режим доступу: https://nure.ua/wp-content/uploads/2018/Scientific_editions/are_2018_16.pdf

ДОДАТОК А ДОДАТКОВІ ФОРМУЛИ

У підрозділі 3.1 показано, що три раунди малоресурсного симетричного блокового шифру «Кипарис» не є стійкими до атак за вибраним відкритим текстом. Для цього застосовувались ідеї Кувакадо Хіденорі та Морія Масакату. Для деякого a отримано представлення у вигляді єдиної формули для $h(a)$ через значення a_0,a_1,a_2,a_3. У даному додатку наведена формула для обчислення $h^2(a)$ через значення a_0,a_1,a_2,a_3. Для цього треба зробити конкатенацію $a^0_{h^2}$,$a^1_{h^2}$,$a^2_{h^2}$,$a^3_{h^2}$, які обчислюються таким чином:

$a^0_{h^2} = (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)) \boxplus ((((a_1 \oplus (a_2 \boxplus \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2) \oplus ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1)) \oplus (((a_3 \oplus (a_0 \boxplus \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)) \lll \lll r_3)) \lll r_4) \boxplus ((((((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2) \oplus ((a_2 \boxplus ((a_3 \oplus \oplus (a_0 \boxplus a_1)) \lll r_1)) \oplus (((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)) \lll r_3)) \lll r_4) \oplus ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1)) \oplus \oplus ((((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll \lll r_2))) \lll r_3) \boxplus ((((((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus \boxplus a_1)) \lll r_1))) \lll r_2)) \lll r_3) \oplus ((a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll \lll r_2)) \boxplus ((((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2) \oplus ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus \boxplus a_1)) \lll r_1)) \oplus (((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus \boxplus a_1)) \lll r_1))) \lll r_2)) \lll r_3)) \lll r_4))) \lll r_1))) \lll r_2)$;

$a^1_{h^2} = (((((((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2) \oplus ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus \boxplus a_1)) \lll r_1)) \oplus (((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus \boxplus a_1)) \lll r_1))) \lll r_2)) \lll r_3)) \lll r_4) \oplus ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1)) \oplus (((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2))) \lll \lll r_3) \boxplus ((((((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll \lll r_1))) \lll r_2))) \lll r_3) \oplus ((a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll \lll r_2)) \boxplus ((((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2) \oplus ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus \boxplus a_1)) \lll r_1)) \oplus (((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus \boxplus a_1)) \lll r_1))) \lll r_2)) \lll r_3)) \lll r_4))) \lll r_1))) \lll r_2) \oplus (((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus$

$$\boxplus a_1)) \lll r_1)) \oplus ((((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus$$
$$\boxplus a_1)) \lll r_1))) \lll r_2))) \lll r_3) \boxplus ((((((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus$$
$$\oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2))) \lll r_3) \oplus ((a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus$$
$$\oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)) \boxplus ((((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2) \oplus$$
$$\oplus ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1)) \oplus (((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus$$
$$\boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)) \lll r_3)) \lll r_4))) \lll r_1)) \oplus (((((((a_3 \oplus (a_0 \boxplus$$
$$\boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2))) \lll r_3) \oplus$$
$$\oplus ((a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)) \boxplus ((((a_1 \oplus (a_2 \boxplus ((a_3 \oplus$$
$$\oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2) \oplus ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1)) \oplus (((a_3 \oplus (a_0 \boxplus$$
$$\boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)) \lll$$
$$\lll r_3)) \lll r_4))) \lll r_1) \oplus ((a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll$$
$$\lll r_2)) \boxplus ((((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2) \oplus ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus$$
$$\boxplus a_1)) \lll r_1)) \oplus (((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus$$
$$\boxplus a_1)) \lll r_1))) \lll r_2)) \lll r_3)) \lll r_4) \boxplus ((((((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll$$
$$\lll r_1))) \lll r_2) \oplus ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1)) \oplus (((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus$$
$$\oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)) \lll r_3)) \lll r_4) \oplus ((a_2 \boxplus$$
$$\boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1)) \oplus ((((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus$$
$$\oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2))) \lll r_3) \boxplus ((((((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus$$
$$\boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2))) \lll r_3) \oplus ((a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus$$
$$\boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)) \boxplus ((((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll$$
$$\lll r_2) \oplus ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1)) \oplus (((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus$$
$$\oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)) \lll r_3)) \lll r_4))) \lll r_1))) \lll r_2)) \lll$$
$$\lll r_3)) \lll r_4;$$

$$a_{h^2}^2 = ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1)) \oplus ((((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus$$
$$\oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2))) \lll r_3) \boxplus ((((((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus$$
$$\oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2))) \lll r_3) \oplus ((a_0 \boxplus a_1 \boxplus$$
$$\boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)) \boxplus ((((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll$$
$$\lll r_1))) \lll r_2) \oplus ((a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1)) \oplus (((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus$$
$$\boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)) \lll r_3)) \lll r_4))) \lll r_1)) \oplus$$
$$\oplus ((((((((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1) \oplus (a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll$$
$$\lll r_2))) \lll r_3) \oplus ((a_0 \boxplus a_1 \boxplus ((a_1 \oplus (a_2 \boxplus ((a_3 \oplus (a_0 \boxplus a_1)) \lll r_1))) \lll r_2)) \boxplus$$

$\boxplus((((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2)\oplus((a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1))\oplus$
$\oplus(((a_3\oplus(a_0\boxplus a_1))\lll r_1)\oplus(a_0\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll$
$\lll r_2))\lll r_3))\lll r_4)))\lll r_1)\oplus((a_0\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll$
$\lll r_1)))\lll r_2))\boxplus((((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2)\oplus((a_2\boxplus((a_3\oplus$
$\oplus(a_0\boxplus a_1))\lll r_1))\oplus(((a_3\oplus(a_0\boxplus a_1))\lll r_1)\oplus(a_0\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus$
$\boxplus a_1))\lll r_1)))\lll r_2))\lll r_3))\lll r_4)\boxplus((((((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll$
$\lll r_1)))\lll r_2)\oplus((a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1))\oplus(((a_3\oplus(a_0\boxplus a_1))\lll r_1)\oplus(a_0\boxplus$
$\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2))\lll r_3))\lll r_4)\oplus((a_2\boxplus((a_3\oplus$
$\oplus(a_0\boxplus a_1))\lll r_1))\oplus((((a_3\oplus(a_0\boxplus a_1))\lll r_1)\oplus(a_0\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus$
$\boxplus a_1))\lll r_1)))\lll r_2)))\lll r_3)\boxplus((((((a_3\oplus(a_0\boxplus a_1))\lll r_1)\oplus(a_0\boxplus a_1\boxplus((a_1\oplus$
$\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2)))\lll r_3)\oplus((a_0\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus((a_3\oplus$
$\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2))\boxplus((((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2)\oplus$
$\oplus((a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1))\oplus(((a_3\oplus(a_0\boxplus a_1))\lll r_1)\oplus(a_0\boxplus a_1\boxplus((a_1\oplus$
$\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2))\lll r_3))\lll r_4)))\lll r_1)))\lll r_2)))\lll$
$\lll r_3)$;

$a^3_{h^2}=(((((((a_3\oplus(a_0\boxplus a_1))\lll r_1)\oplus(a_0\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus((a_3\lll r_2)))\lll r_3)\oplus$
$\oplus((a_0\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2))\boxplus((((a_1\oplus(a_2\boxplus((a_3\oplus$
$\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2)\oplus((a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1))\oplus(((a_3\oplus(a_0\boxplus$
$\boxplus a_1))\lll r_1)\oplus(a_0\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2))\lll r_3))\lll$
$\lll r_4)))\lll r_1)\oplus((a_0\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2))\boxplus$
$\boxplus((((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2)\oplus((a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll$
$\lll r_1))\oplus(((a_3\oplus(a_0\boxplus a_1))\lll r_1)\oplus(a_0\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll$
$\lll r_1)))\lll r_2))\lll r_3))\lll r_4)\boxplus((((((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll$
$\lll r_2)\oplus((a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1))\oplus(((a_3\oplus(a_0\boxplus a_1))\lll r_1)\oplus(a_0\boxplus a_1\boxplus$
$\boxplus((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2))\lll r_3))\lll r_4)\oplus((a_2\boxplus((a_3\oplus$
$\oplus(a_0\boxplus a_1))\lll r_1))\oplus((((a_3\oplus(a_0\boxplus a_1))\lll r_1)\oplus(a_0\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus$
$\boxplus a_1))\lll r_1)))\lll r_2)))\lll r_3)\boxplus((((((a_3\oplus(a_0\boxplus a_1))\lll r_1)\oplus(a_0\boxplus a_1\boxplus((a_1\oplus$
$\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2)))\lll r_3)\oplus((a_0\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus((a_3\oplus$
$\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2))\boxplus((((a_1\oplus(a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2)\oplus$
$\oplus((a_2\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1))\oplus(((a_3\oplus(a_0\boxplus a_1))\lll r_1)\oplus(a_0\boxplus a_1\boxplus((a_1\oplus(a_2\boxplus$
$\boxplus((a_3\oplus(a_0\boxplus a_1))\lll r_1)))\lll r_2))\lll r_3))\lll r_4)))\lll r_1)))\lll r_2)))\lll r_3.$

ДОДАТОК Б ПРОГРАМНІ ОБРАХУНКИ ЛОГАРИФМА ЙМОВІРНОСТІ РОТАЦІЇ

У підрозділі 2.2 наведена таблиця значень двійкових логарифмів ймовірностей ротації для послідовних додавань 64-бітових слів, значення яких взяті зі статті [28] (див. таблицю 2.1). Однак для роботи з 32-бітовими словами така таблиця не наводилась, тому на практиці була реалізована програма на мові Python, яка, використовуючи модуль Decimal для точності результатів та бібліотеку SciPy для обчислення значень біноміальних коефіцієнтів, зробила обрахунки на такий випадок. Виявилось, що значення двійкових логарифмів ймовірностей для 32- та 64-бітових слів різняться щонайбільше на одну мільйонну.

У таблиці Б.1 наведені результати роботи програми тільки для 112 можливих додавань, адже для ротаційного криптоаналізу шифру «Кипарис» у підрозділі 3.3 більше і не треба.

Таблиця Б.1 – Значення двійкових логарифмів ймовірностей ротації для послідовних додавань за теоремою 2.4

Параметр ротації $r = 1$				
Кількість додавань	1	2	3	4
для 32-бітових	-1.415037499	-3.584962499	-6.263034402	-9.321928088
для 64-бітових	-1.415037499	-3.584962501	-6.263034406	-9.321928095
Кількість додавань	5	6	7	8
для 32-бітових	-12.684498164	-16.299208004	-20.129282998	-24.147204901
для 64-бітових	-12.684498174	-16.299208018	-20.129283017	-24.147204925
Кількість додавань	9	10	11	12
для 32-бітових	-28.331629466	-32.665530196	-37.135015472	-41.728539978
для 64-бітових	-28.331629496	-32.665530233	-37.135015516	-41.728540030
Кількість додавань	13	14	15	16
для 32-бітових	-46.436359218	-51.250140399	-56.162677548	-61.167678218
для 64-бітових	-46.436359279	-51.250140470	-56.162677629	-61.167678310
Кількість додавань	17	18	19	20
для 32-бітових	-66.259600696	-71.433527616	-76.685066370	-82.010269584
для 64-бітових	-66.259600799	-71.433527731	-76.685066498	-82.010269725

Параметр ротації $r = 1$				
Кількість додавань	21	22	23	24
для 32-бітових	-87.405570851	-92.867732248	-98.393801044	-103.98107369
для 64-бітових	-87.405571006	-92.867732418	-98.393801229	-103.98107389
Кількість додавань	25	26	27	28
для 32-бітових	-109.62706561	-115.32948567	-121.08621450	-126.89528588
для 64-бітових	-109.62706582	-115.32948591	-121.08621476	-126.89528615
Кількість додавань	29	30	31	32
для 32-бітових	-132.75487074	-138.66326334	-144.61886920	-150.62019458
для 64-бітових	-132.75487103	-138.66326365	-144.61886953	-150.62019493
Кількість додавань	33	34	35	36
для 32-бітових	-156.66583722	-162.75447823	-168.88487484	-175.05585404
для 64-бітових	-156.66583760	-162.75447863	-168.88487527	-175.05585448
Кількість додавань	37	38	39	40
для 32-бітових	-181.26630682	-187.51518314	-193.80148730	-200.12427386
для 64-бітових	-181.26630729	-187.51518363	-193.80148782	-200.12427441
Кількість додавань	41	42	43	44
для 32-бітових	-206.48264392	-212.87574178	-219.30275189	-225.76289610
для 64-бітових	-206.48264450	-212.87574239	-219.30275253	-225.76289677
Кількість додавань	45	46	47	48
для 32-бітових	-232.25543113	-238.77964630	-245.33486143	-251.92042489
для 64-бітових	-232.25543183	-238.77964703	-245.33486219	-251.92042568
Кількість додавань	49	50	51	52
для 32-бітових	-258.53571190	-265.18012283	-271.85308178	-278.55403515
для 64-бітових	-258.53571272	-265.18012369	-271.85308267	-278.55403608
Кількість додавань	53	54	55	56
для 32-бітових	-285.28245041	-292.03781487	-298.81963467	-305.62743366
для 64-бітових	-285.28245137	-292.03781587	-298.81963570	-305.62743474
Кількість додавань	57	58	59	60
для 32-бітових	-312.46075257	-319.31914803	-326.20219184	-333.10947017
для 64-бітових	-312.46075368	-319.31914918	-326.20219303	-333.10947140
Кількість додавань	61	62	63	64
для 32-бітових	-340.04058282	346.99514263	-353.97277477	-360.97311624
для 64-бітових	-340.04058409	-346.99514394	-353.97277613	-360.97311764
Кількість додавань	65	66	67	68
для 32-бітових	-367.99581524	-375.04053074	-382.10693192	-389.19469777
для 64-бітових	-367.99581668	-375.04053222	-382.10693345	-389.19469935
Кількість додавань	69	70	71	72
для 32-бітових	-396.30351664	-403.43308583	-410.58311122	-417.75330693
для 64-бітових	-396.30351826	-403.43308750	-410.58311294	-417.75330869
Кількість додавань	73	74	75	76
для 32-бітових	-424.94339492	-432.15310474	-439.38217317	-446.63034399
для 64-бітових	-424.94339673	-432.15310660	-439.38217509	-446.63034595

Параметр ротації $r = 1$				
Кількість додавань	77	78	79	80
для 32-бітових	-453.89736762	-461.18300097	-468.48700711	-475.80915505
для 64-бітових	-453.89736964	-461.18300304	-468.48700923	-475.80915723
Кількість додавань	81	82	83	84
для 32-бітових	-483.14921958	-490.50698096	-497.88222482	-505.27474188
для 64-бітових	-483.14922181	-490.50698325	-497.88222716	-505.27474427
Кількість додавань	85	86	87	88
для 32-бітових	-512.68432783	-520.11078315	-527.55391290	-535.01352660
для 64-бітових	-512.68433029	-520.11078566	-527.55391547	-535.01352923
Кількість додавань	89	90	91	92
для 32-бітових	-542.48943810	-549.98146536	-557.48943040	-565.01315911
для 64-бітових	-542.48944079	-549.98146811	-557.48943321	-565.01316198
Кількість додавань	93	94	95	96
для 32-бітових	-572.55248114	-580.10722979	-587.67724189	-595.26235766
для 64-бітових	-572.55248408	-580.10723279	-587.67724495	-595.26236079
Кількість додавань	97	98	99	100
для 32-бітових	-602.86242067	-610.47727765	-618.10677848	-625.75077604
для 64-бітових	-602.86242386	-610.47728091	-618.10678181	-625.75077943
Кількість додавань	101	102	103	104
для 32-бітових	-633.40912613	-641.08168739	-648.76832124	-656.46889176
для 64-бітових	-633.40912959	-641.08169092	-648.76832484	-656.46889542
Кількість додавань	105	106	107	108
для 32-бітових	-664.18326561	-671.91131201	-679.65290261	-687.40791148
для 64-бітових	-664.18326935	-671.91131582	-679.65290650	-687.40791543
Кількість додавань	109	110	111	112
для 32-бітових	-695.17621497	-702.95769170	-710.75222251	-718.55969034
для 64-бітових	-695.17621899	-702.95769580	-710.75222669	-718.55969460
Параметр ротації $r = 2$				
Кількість додавань	1	2	3	4
для 32-бітових	-1.678071904	-4.263034402	-7.455679476	-11.099535660
для 64-бітових	-1.678071905	-4.263034406	-7.455679484	-11.099535674
Кількість додавань	5	6	7	8
для 32-бітових	-15.099535653	-19.392317395	-23.932885767	-28.687773258
для 64-бітових	-15.099535674	-19.392317423	-23.932885804	-28.687773306
Кількість додавань	9	10	11	12
для 32-бітових	-33.631189717	-38.742698019	-44.005732410	-49.406611830
для 64-бітових	-33.631189778	-38.742698093	-44.005732499	-49.406611935
Кількість додавань	13	14	15	16
для 32-бітових	-54.933858816	-60.577714987	-66.329787453	-72.182785019
для 64-бітових	-54.933858938	-60.577715128	-66.329787614	-72.182785202
Кількість додавань	17	18	19	20
для 32-бітових	-78.130317577	-84.166740961	-90.287035169	-96.486707487
для 64-бітових	-78.130317782	-84.166741190	-90.287035424	-96.486707769

Параметр ротації $r = 2$				
Кількість додавань	21	22	23	24
для 32-бітових	-102.76171451	-109.10839867	-115.52343614	-122.00379356
для 64-бітових	-102.76171482	-109.10839901	-115.52343651	-122.00379397
Кількість додавань	25	26	27	28
для 32-бітових	-128.54669197	-135.14957635	-141.81008984	-148.52605180
для 64-бітових	-128.54669241	-135.14957682	-141.81009035	-148.52605234
Кількість додавань	29	30	31	32
для 32-бітових	-155.29543883	-162.11636857	-168.98708551	-175.90594870
для 64-бітових	-155.29543941	-162.11636919	-168.98708618	-175.90594941
Кількість додавань	33	34	35	36
для 32-бітових	-182.87142098	-189.88205945	-196.93650719	-204.03348578
для 64-бітових	-182.87142173	-189.88206025	-196.93650803	-204.03348667
Кількість додавань	37	38	39	40
для 32-бітових	-211.17178875	-218.35027571	-225.56786709	-232.82353943
для 64-бітових	-211.17178969	-218.35027671	-225.56786814	-232.82354053
Кількість додавань	41	42	43	44
для 32-бітових	-240.11632113	-247.44528862	-254.80956295	-262.20830658
для 64-бітових	-240.11632228	-247.44528984	-254.80956422	-262.20830791
Кількість додавань	45	46	47	48
для 32-бітових	-269.64072059	-277.10604204	-284.60354164	-292.13252154
для 64-бітових	-269.64072198	-277.10604350	-284.60354315	-292.13252312
Кількість додавань	49	50	51	52
для 32-бітових	-299.69231340	-307.28227652	-314.90179617	-322.55028209
для 64-бітових	-299.69231505	-307.28227823	-314.90179795	-322.55028394
Кількість додавань	53	54	55	56
для 32-бітових	-330.22716701	-337.93190537	-345.66397209	-353.42286145
для 64-бітових	-330.22716893	-337.93190736	-345.66397416	-353.42286359
Кількість додавань	57	58	59	60
для 32-бітових	-361.20808602	-369.01917573	-376.85567692	-384.71715152
для 64-бітових	-361.20808824	-369.01917803	-376.85567930	-384.71715397
Кількість додавань	61	62	63	64
для 32-бітових	-392.60317624	-400.51334189	-408.44725261	-416.40452531
для 64-бітових	-392.60317878	-400.51334451	-408.44725532	-416.40452810
Кількість додавань	65	66	67	68
для 32-бітових	-424.38478900	-432.38768428	-440.41286275	-448.45998657
для 64-бітових	-424.38479189	-432.38768725	-440.41286581	-448.45998973
Кількість додавань	69	70	71	72
для 32-бітових	-456.52872796	-464.61876873	-472.72979995	-480.86152146
для 64-бітових	-456.52873120	-464.61877207	-472.72980339	-480.86152499
Кількість додавань	73	74	75	76
для 32-бітових	-489.01364155	-497.18587661	-505.37795079	-513.58959568
для 64-бітових	-489.01364518	-497.18588034	-505.37795462	-513.58959961

Параметр ротації $r = 2$				
Кількість додавань	77	78	79	80
для 32-бітових	-521.82055001	-530.07055940	-538.33937605	-546.62675852
для 64-бітових	-521.82055404	-530.07056354	-538.33938029	-546.62676288
Кількість додавань	81	82	83	84
для 32-бітових	-554.93247149	-563.25628549	-571.59797672	-579.95732686
для 64-бітових	-554.93247595	-563.25629006	-571.59798141	-579.95733166
Кількість додавань	85	86	87	88
для 32-бітових	-588.33412283	-596.72815661	-605.13922509	-613.56712988
для 64-бітових	-588.33412774	-596.72816163	-605.13923023	-613.56713514
Кількість додавань	89	90	91	92
для 32-бітових	-622.01167714	-630.47267745	-638.94994563	-647.44330063
для 64-бітових	-622.01168252	-630.47268295	-638.94995125	-647.44330637
Кількість додавань	93	94	95	96
для 32-бітових	-655.95256536	-664.47756661	-673.01813486	-681.57410423
для 64-бітових	-655.95257123	-664.47757261	-673.01814099	-681.57411048
Кількість додавань	97	98	99	100
для 32-бітових	-690.14531230	-698.73160007	-707.33281179	-715.94879490
для 64-бітових	-690.14531869	-698.73160659	-707.33281844	-715.94880169
Кількість додавань	101	102	103	104
для 32-бітових	-724.57939993	-733.22448039	-741.88389271	-750.55749610
для 64-бітових	-724.57940685	-733.22448745	-741.88389990	-750.55750344
Кількість додавань	105	106	107	108
для 32-бітових	-759.24515254	-767.94672666	-776.66208565	-785.39109924
для 64-бітових	-759.24516002	-767.94673428	-776.66209342	-785.39110714
Кількість додавань	109	110	111	112
для 32-бітових	-794.13363955	-802.88958112	-811.65880077	-820.44117755
для 64-бітових	-794.13364761	-802.88958933	-811.65880912	-820.44118605