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

«На правах рукопису»
УДК 004.056.55:512.6

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

Дипломна робота

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

зі спеціальності: 113 Прикладна математика
на тему: **«Оцінювання стійкості ускладнюючих функцій спеціального виду до обертального криптоаналізу»**

Виконав:
студент II курсу, групи ФІ-03
Бондар Петро Олександович _____

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

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

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

Студент _____

Київ — 2024

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

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

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

ЗАВДАННЯ
на дипломну роботу

Студент: Бондар Петро Олександович

1. Тема роботи: «*Оцінювання стійкості ускладнюючих функцій спеціального виду до обертального криптоаналізу*», науковий керівник дисертації: доцент кафедри ММЗІ, к.т.н. Яковлєв Сергій Володимирович,

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

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

3. Об'єкт дослідження: інформаційні процеси в системах захисту інформації.

4. Предмет дослідження: методи обертального криптоаналізу ARX-криптосистем.

5. Перелік завдань:

1) провести огляд опублікованих джерел за тематикою дослідження;

2) знайти імовірності проходження довільних пар обертання через функцію множення на три для векторів довільної довжини;

3) знайти імовірності проходження деяких пар обертання через функцію множення на п'ять для векторів довільної довжини;

4) перевірити одержані результати експериментально.

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

7. Орієнтовний перелік публікацій: публікація на XXII Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» у секції «Актуальні проблеми криптографічного захисту інформації».

8. Дата видачі завдання: 10 вересня 2023 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	01-15 вересня 2023 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	Вересень - жовтень 2023 р.	Виконано
2	Ознайомлення із попередніми застосуваннями методу обертального криптоаналізу та аналіз доведень	Листопад-грудень 2023 р.	Виконано
3	Побудова оцінок імовірності для функції множення на три	Січень-лютий 2024 р.	Виконано
4	Побудова оцінок імовірності для функції множення на п'ять	Березень-квітень 2024 р.	Виконано
5	Розробка програмного засобу для перевірки одержаних результатів та підтвердження результатів	Квітень 2024 р.	Виконано
6	Оформлення дипломної роботи	Травень 2024 р.	Виконано

Студент _____ Петро БОНДАР

Керівник _____ Сергій ЯКОВЛЄВ

РЕФЕРАТ

Кваліфікаційна робота містить: 54 сторінки, 11 рисунків, 13 таблиць, 11 джерел.

У цій роботі наведено огляд поточних результатів дослідження та застосування методу обертального криптоаналізу, а також застосування цього методу до окремих ускладнюючих функцій, що були використині у геш-функції Shabal. Основним предметом досліження є функції $f_1(x) = 3x \bmod 2^n$ та $f_2(x) = 5x \bmod 2^n$.

В ході дослідження побудовано оцінки ймовірностей проходження пар обертання через функцію $f_1(x)$ для довільних значень довжини бітових векторів та для довільного значення обертання. А також, оцінки для функції $f_2(x)$ для обертаннь на 1 та $n - 1$. Отримані результати проаналізовано та перевірено на практиці шляхом порівняння практичних результатів та результатів, отриманих за допомогою сформульованих оцінок.

ОБЕРТАЛЬНИЙ КРИПТОАНАЛІЗ, ARX-КРИПТОСИСТЕМИ, ГЕШ-ФУНКЦІЯ SHABAL

ABSTRACT

The qualification work contains: 54 pages, 11 figures, 13 tables, 11 citations.

This work presents an overview of the current research results and application of the rotational cryptoanalysis, and the application of this method to individual complicating functions used in the Shabal hash function. The main research subjects are complicating functions $f_1(x) = 3x \bmod 2^n$ and $f_2(x) = 5x \bmod 2^n$.

During the research, estimates of the probabilities of rotational pairs transition through the function $f_1(x)$ for arbitrary bit vector lengths and rotation values were constructed. Also, probability formulas for the function $f_2(x)$ were constructed for rotation values 1 and $n - 1$. The results were analyzed and verified by comparing the practical results with those obtained using the formulated estimates.

ROTATIONAL CRYPTOANALYSIS, ARX CRYPTOSYSTEMS, SHABAL HASH FUNCTION

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ARX — Addition, Rotation, XOR (Додавання, обертання, XOR).

$V_n = \{0,1\}^n$ — множина двійкових векторів довжини n.

$\mathcal{F} = \{f : V_n^m \to V_n \mid m \in \mathbb{N}\}$ — множина всіх функцій на V_n.

$x = (x_{n-1}, x_{n-2}, \ldots, x_0) \in V_n$ — двійковий вектор довжини n.

$\lll r$ / $\ggg r$ — операція циклічного зсуву вліво/вправо на r бітів.

x^r / x^{-r} — бітовий вектор x, циклічно зсунутий вліво/вправо на r бітів ($x \lll r$ / $x \ggg r$).

$\oplus$ — операція побітового додавання (XOR).

$\boxplus$ — операція додавання за модулем 2^n ($x \boxplus y = (x + y) \bmod 2^n$).

$\langle a, b, c \rangle$ — функція мажоризації, що набуває значення 1, якщо хоча б два вхідних значення дорівнюють 1, інакше – набуває значення 0.

C_n^k — біноміальний коефіцієнт, значення якого обчислюється як $\frac{n!}{k!(n-k)!}$.

ВСТУП

Актуальність дослідження. Актуальність даного дослідження полягає у значній поширеності використання ARX-криптосистем на практиці, особливо в системах з обмеженими ресурсами, завдяки простоті реалізації таких криптосистем та ефективності виконання операцій, на яких вони ґрунтуються. Метод обертального криптоаналізу найкраще підходить для аналізу криптосистем саме такого типу, але при цьому він є відносно новим, через що виникає окремий інтерес до нього.

Метою дослідження є уточнення та покращення методів криптоаналізу ARX-криптосистем. Для досягнення мети необхідно вирішити такі завдання дослідження:

1) провести огляд опублікованих джерел за тематикою дослідження;

2) знайти імовірності проходження довільних пар обертання через функцію множення на три для векторів довільної довжини;

3) знайти імовірності проходження певних пар обертання через функцію множення на п'ять для векторів довільної довжини;

4) перевірити одержані результати експериментально.

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

Предметом дослідження є методи обертального криптоаналізу та його застосування до ARX-криптосистем.

При розв'язанні поставлених завдань використовувались такі *методи дослідження*: дискретної математики, теорії імовірності, комбінаторного аналізу та комп'ютерного моделювання.

Наукова новизна отриманих результатів полягає в тому, що вперше було одержано аналітичні вирази для імовірностей проходження пар обертання для функцій множення на три, для довільної довжини вектора та довільного значення обертання, та множення на п'ять, для довільної довжини вектора та окремих значень обертання.

Практичне значення результатів полягає в тому, що одержані результати дозволяють будувати нові надійні ARX-криптосистеми.

Апробація результатів та публікації. Результати цієї роботи були опубліковані на XXII Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» у секції «Актуальні проблеми криптографічного захисту інформації» [1].

1 ОБЕРТАЛЬНИЙ КРИПТОАНАЛІЗ ARX-КРИПТОСИСТЕМ

В першому розділі розглянуті поняття ARX-криптосистеми та обертального криптоаналізу, приклади застосування цього методу криптоаналізу на деякі окремі криптопримітиви або результати його застосування. Крім того, в цьому розділі буде розглянуто основні результати аналізу, що були отримані в попередніх роботах, функцій, які будуть розглянуті у другому розділі цієї роботи – $f_1(x) = 3x \bmod 2^n$ та $f_2(x) = 5x \bmod 2^n$.

1.1 ARX-криптосистеми та ARX-C-криптосистеми

Сформулюємо основні означення і твердження, що стосуються ARX-криптосистем.

Означення 1.1. Криптосистеми, що ґрунтуються на операціях модульного додавання, циклічних зсувів та побітового виключного АБО, називаються **ARX-криптосистемами** (*англ.* Addition, Rotation, XOR).

Внаслідок простоти реалізації та ефективності виконання зазначених операцій, системи такого вигляду активно використовуються на практиці.

Приклад 1.1. Прикладами таких систем є:
– геш-функції Skein, BLAKE, CubeHash, Keccak;
– потокові шифри сімейства Salsa20;
– блокові шифри TEA, XTEA, Speck;
– MAC алгоритм Chaskey.

Класичним методом ускладнення криптоаналізу системи є додавання констант у внутрішню структуру примітиву.

Означення 1.2. Криптосистеми, що ґрунтуються на операціях

модульного додавання, циклічних зсувів, побітового виключного АБО та використанні констант, називаються **ARX-C-криптосистемами** (*англ.* Addition, Rotation, XOR, Constants).

Ховратовичем та Ніколічем у їх роботі [2] 2010 року була сформульована і доведена теорема, що стверджує про можливість реалізації довільної функції над V_n за допомогою такого набору операцій.

Теорема 1.1 (Повнота ARX-C [2])**.** *Набір функцій* $\{\boxplus, \oplus, \ggg 1, 1\}$ *є базисом множини* $\mathcal{F}$*.*

Щобільше, в цій роботі було показано, що з цього набору операцій можливо прибрати операцію XOR. Але при використанні тільки додавання за модулем і циклічного зсуву функцію можна наблизити до лінійної, через що суттєво знижується її стійкість.

В загальному випадку ніщо не заважає замінити в цьому наборі операцій циклічний зсув праворуч ($\ggg$) на циклічний зсув ліворуч ($\lll$), бо вони дуже просто виражаються один через одного.

1.2 Обертальний криптоаналіз

Метод обертального криптоаналізу, хоч на той момент він так не називався, був застосований ще у 2006 році розробниками блокового шифру SEA [3]. Також, вони продемонстрували спосіб захисту від такого типу атак за допомогою використання нелінійного генератора ключів та псевдовипадкових констант.

Але вперше формалізований даний метод був у статті Ховратовича та Ніколіча [2]. Для цього було введено поняття так званих пар обертання. Для зручності використання цього означення надалі в цій роботі, розглядатимемо зсув ліворуч в контексті пар обертання.

Означення 1.3. Парою обертання (з обертанням r) називається

довільна пара векторів вигляду (x, x^r), тобто:

$$x = (x_{n-1}, x_{n-2}, \ldots, x_k \ldots, x_0) \Leftrightarrow$$
$$\Leftrightarrow x^r = (x_{n-r-1}, x_{n-r-2}, \ldots, x_{k-r \, mod n} \ldots, x_{n-r}).$$

Тоді, обертальний криптоаналіз визначається таким чином.

Означення 1.4. Обертальний криптоаналіз – це ймовірнісна атака, що ґрунтується на дослідженні змін в парах обертання при проходженні через раундову функцію або систему.

При застосуванні обертального криптоаналізу стійкість системи оцінюється за ймовірністю проходження пар обертання через криптопримітив чи систему.

Ймовірність проходження пари обертання через відображення (систему) $f : V_n \to V_n$ визначається так:

$$rp^f(r) = \Pr\{f(x^r) = (f(x))^r\}.$$

В межах цієї роботи вважатимемо, що розподіл векторів $x \in V_n$ є рівноймовірним. Інакше кажучи, кожен вектор $x \in V_n$ має однакову ймовірність появи $\frac{1}{2^n}$.

В тій самій статті [2] автори сформулювали низку базових результатів, що поклали початок дослідженням такого типу атак.

Твердження 1.1 (Ховратович, Ніколіч [2])**.** *Операції XOR та циклічного зсуву зберігають проходження пар обертання:*

$$(x \oplus y)^r = x^r \oplus x^r;$$
$$(x \lll k)^r = x^r \lll k.$$

Для додавання за модулем 2^n виявилось не все так просто. Даум у 2005 році у своїй докторській дисертації [4] сформулював оцінку для збереження обертальності для пари після проходження через цю операцію.

Твердження 1.2 (Даум [4])**.** *Ймовірність проходження пари*

обертання через операцію $\boxplus$ *можна обчислити таким чином:*

$$\Pr\{(x \boxplus y)^r = x^r \boxplus y^r\} = \frac{1}{4}(1 + 2^{r-n} + 2^{-r} + 2^{-n}).$$

Таким чином, було показано, що ця ймовірність залежить лише від значення обертання r та кількості бітів у словах n.

Враховуючи вище наведені твердження, Ховратович та Ніколіч побудували оцінку для проходження пари обертання через певну ARX-схему.

Твердження 1.3 (Ховратович, Ніколіч [2])**.** *Нехай* $\mathcal{S}$ *– певна ARX схема, що містить в собі рівно* q *операцій додавання за модулем, а* p_r *– ймовірність проходження обертальної пари через операцію* $\boxplus$*.*

В такому випадку, для певного входу I*, який утворює обертальну пару* (I, I^r)*, маємо таку оцінку ймовірності проходження цієї обертальної пари:*

$$\Pr\{\mathcal{S}(I^r) = (\mathcal{S}(I))^r\} = (p_r)^q.$$

Але така оцінка в результаті виявилась не точною через те, що вона була побудована за припущення того, що події проходження пар обертання через операцію додавання за модулем утворюють ланцюг Маркова. Це в загальному випадку виявилось неправильним.

Обертальний криптоаналіз в ARX-C-криптосистемах

В деяких криптосистемах з ітеративною структурою задля запобігання атак зсувів використовуються різні раундові константи.

Для адаптації обертального криптоаналізу під таке використання констант було введено поняття обертальної похибки.

Означення 1.5. Обертальна похибка E (при обертанні r) для x та y визначається так:

$$E(x, y) = x^r \oplus y.$$

Майже очевидним є те, що для пари обертання (x, x^r) виконується рівність $E(x, x^r) = 0$.

Власне обертальну похибку може спричинити як константа в системі, так і операція модульного додавання. Але це не є обов'язковим, оскільки через них же, ці похибки можуть компенсувати одна одну з певною імовірністю.

Ховратович та Ніколіч у статті [2] також сформулювали таке твердження.

Твердження 1.4. *Якщо C – певна константа, що використовується в схемі, тоді чим менша відстань Геммінга між C та C^r, тим більша ймовірність проходження обертальної пари через цю схему.*

Також стверджувалось, що найкраще обертальний криптоаналіз працює на схемах, що використовують інваріантні чи майже інваріантні відносно обертання константи.

1.3 Покращення формалізації обертального криптоаналізу

Для диференціального криптоаналізу було сформульовано означення марковості шифру.

Означення 1.6. Шифр з ітеративною структурою, що має раундову функцію $f(x, z) = y$, називається **марковим**, якщо для довільних $\alpha, \beta \neq 0$ виконується

$$\forall \gamma \ : \ \Pr\{\Delta y = \beta \mid \Delta x = \alpha, x = \gamma\} = \Pr\{\Delta y(1) = \beta_1 \mid \Delta x = \alpha\},$$

за умови, що раундові ключі z обираються незалежно та рівноймовірно. Тут Δ – позначає диференціал на вході чи виході.

Після формалізації обертального криптоаналізу було помічено, що припущення марковості шифру, яке призводило до припущення ланцюга

Маркова (незалежності переходів), не виконується в загальному випадку для проходження обертальних пар. Спростування цього було наведено в іншій роботі Ховратовича та Ніколіча [5]. В цій самій роботі було показано, що ймовірність проходження пари обертання через криптопримітив залежить не тільки від кількості додавань, як це формулювалось у твердженні 1.3, а ще й від їх взаємного розташування у схемі.

Автори навели приклад двох ARX-систем (рис. 1.1), що мають однакову кількість додавань за модулем, але при цьому мають різні ймовірності проходження обертальних пар.

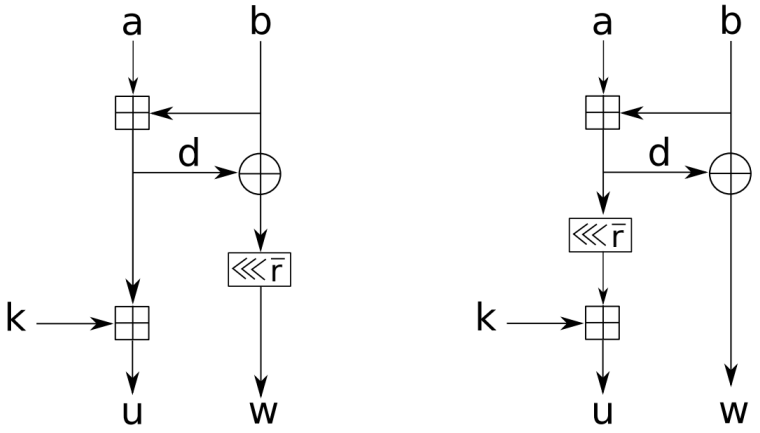

Рисунок 1.1 – Приклад двох ARX криптопримітивів з однаковою кількістю додавань, але різною ймовірністю проходження пар обертання.

Було сформульовано уточнену оцінку для цієї імовірності, яка враховує не тільки послідовні додавання, а весь ланцюг додавань у схемі.

Теорема 1.2 (Про ланцюгові додавання за модулем [5])**.** *Якщо* $a_1, \ldots, a_k$ *– випадково обрані слова з* V_n*, в такому випадку виконується:*

$$\Pr\left\{\bigwedge_{i=2}^{k}[(a_1 \boxplus \cdots \boxplus a_i)^r = a_1^r \boxplus \cdots \boxplus a_i^r]\right\} = \frac{1}{2^{nk}} C_{k+2^r-1}^{2^r-1} C_{k+2^{n-r}-1}^{2^{n-r}-1}$$

Напряму з цієї теореми випливає твердження 1.5.

Твердження 1.5 (Ховратович, Ніколіч [5]). *Ланцюгові додавання за модулем не формують ланцюг Маркова відносно обертальних різниць (обертальних похибок).*

Отже, ймовірності проходження пар обертання через ланцюг додавань за модулем не можуть бути пораховані як добуток ймовірностей проходження через одне додавання за модулем, що спростовує твердження 1.3.

Ховратович та Ніколіч також побудували порівняльну таблицю для оцінки за твердженням 1.3 та оцінки ймовірностей за теоремою 1.2. Отримані результати наведені у таблиці Б.1.

Обертальний криптоаналіз за наявності констант в станах

Також важливою була робота Ашура та Лю [6], які дослідили вплив додавання констант у стани системи на обертальний криптоаналіз. У цій статті вони запропонували новий спосіб обчислення ймовірностей проходження обертальних пар, до яких додаються константи. Таким чином, вони поєднали дві техніки: обертальний та диференціальний криптоаналіз. Для цього автори ввели поняття RX-диференціалу та низку допоміжних тверджень та теорем.

Означення 1.7 ([6]). Нехай (a_1, a_2) утворюють пару обертання. Тоді $((a_1, a_2), r)$**-RX-диференціалом** називається пара обертання (з обертанням r) з перетвореннями a_1 та a_2, тобто $(x \oplus a_1, \overrightarrow{x} \oplus a_2)$. Така пара також називається **RX-парою**.

Результати їх дослідження були експериментально підтверджені за допомогою побудови і аналізу розпізнавача для SPECK32/64 із 7-ми раундовим перетворенням, побудованого на RX-диференціалах.

1.4 Приклади використання обертального криптоаналізу

Застосування обертального криптоаналізу: Skein/Threefish

В статті [2] було вперше застосовано Rebound-атаку, що являє собою поєднання диференціального та обертального криптоаналізу до кандидата конкурсу SHA-3, геш-функції Skein.

Skein – це сімейство геш-функцій, що у своїй основі використовують блоковий шифр Threefish. На конкурс SHA-3 були запропоновані функції на основі Threefish-256 (256-бітовий блок та 256-бітовий ключ) та Threefish-512 (512-бітові блок та ключ).

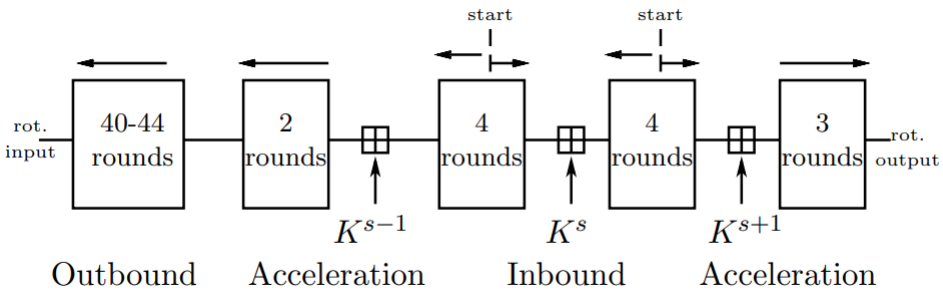

Рисунок 1.2 – Повний опис побудованої атаки на Threefish-256 та Threefish-512. Стрілки позначають напрямок обчислень

Порівняння побудованої атаки з іншими атаками можна побачити в таблиці 1.1. Нехай побудована атака і не загрожує практично застосованим Skein та Threefish з повною кількістю раундів, проте було показано, що за умови зафіксованості більшості або всіх значень такі системи поводяться не випадково. Автори також стверджують, що зменшення кількості раундів у шифрі Threefish до приблизно 53 чи 57 раундів, то модель з вибраним ключем буде поводити себе не випадково у контексті обертальних властивостей.

Таблиця 1.1 – Порівняння атак на Skein-256/512

Раунди	Атака	Метод
Skein/Threefish-256 (72 раунди)		
24*	Відновлення ключа	Диференціальна атака на пов'язаних ключах
39	Відновлення ключа	Обертальна атака на пов'язаних ключах
53	Обертальні колізії	Rebound-атака з обертанням
Skein/Threefish-512 (72 раунди)		
25*	Відновлення ключа	Диференціальна атака на пов'язаних ключах
33*	Відновлення ключа	Бумерангова атака на пов'язаних ключах
35*	Відновлення ключа	Атака розпізнавання на відомих пов'язаних ключах
42	Відновлення ключа	Обертальна атака на пов'язаних ключах
57	Обертальні колізії	Rebound-атака з обертанням

Застосування обертального криптоаналізу: KECCAK

В цій праці [7] було побудовано атаку на ще одного кандидата з конкурсу SHA-3 – геш-функції KECCAK зі зменшеною кількістю раундів.

KECCAK – це геш-функція що має Sponge-будову (рис. 1.3). Криптоаналіз будувався на варіанті функції з розміром стану 1600 бітів. Стандартні значення рейту та ємності становлять $r = 1024$ та $c = 512$ відповідно.

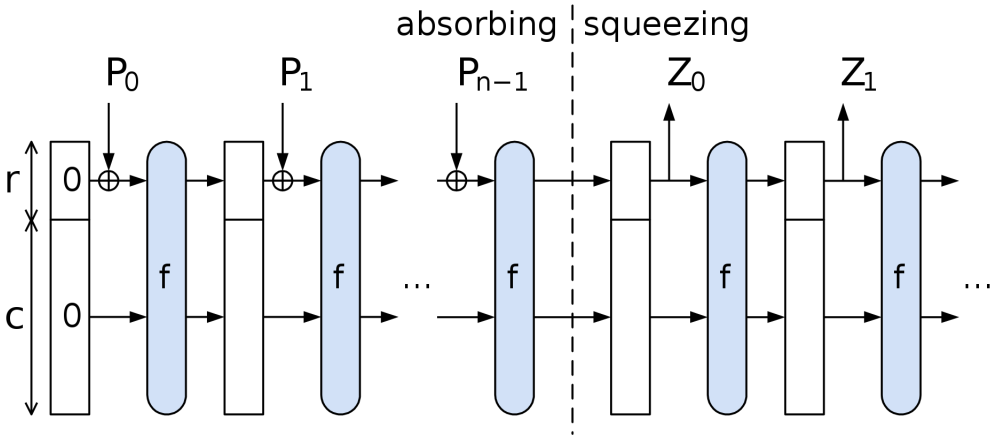

Рисунок 1.3 – Sponge будова для геш-функцій (використовується у Keccak).

Власне розмір стану визначає кількість раундів у функції Keccak-f. В класичному варіанті з розміром стану 1600 (Keccak-f[1600]) бітів перетворення складається з 24 раундів. Раунди відрізняються тільки константами.

В результаті дослідження авторам вдалося досягти наступних результатів:

– **Атака пошуку прообразу** на Keccak із 4-раундовим перетворенням зі складністю 2^{506}.

– **Розпізнавач** для 5-раундового перетворення Keccak-f[1600] зі складністю 2^{29}.

Застосування обертального криптоаналізу: Chaskey

В цій роботі [8] 2020 року обертальний криптоаналіз був застосований для MAC алгоритму Chaskey, що був розроблений для 32-бітних мікроконтролерів.

На рисунках 1.4 та 1.5 можна побачити алгоритм побудови MAC-коду та один раунд перетворення для алгоритму Chaskey.

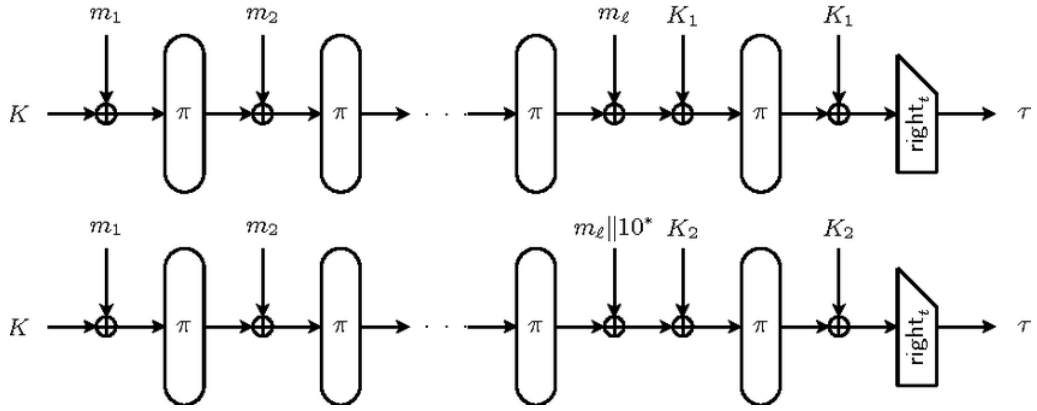

Рисунок 1.4 – Будова MAC-алгоритму Chaskey.

В результаті цієї роботи було побудовано атаку розпізнавання на повну кількість раундів алгоритму зі складністю 2^{86}.

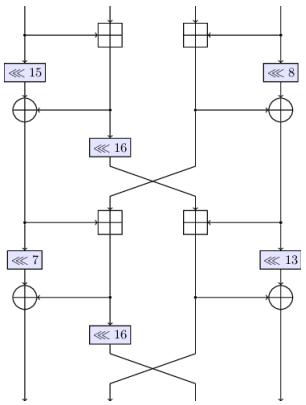

Рисунок 1.5 – Один раунд перетворення Chaskey.

Застосування обертального криптоаналізу: Shabal

В статті [9] було застосовано розпізнавач, побудований з використанням обертального криптоаналізу, на перетворення з геш-функції Shabal, яка також була кандидатом конкурсу SHA-3.

Таке перетворення приймає на вхід параметри $M \in V_{512}$, $A \in V_{384}$, $B \in V_{512}$, $C \in V_{512}$, ці параметри можна також розглядати як конкатенацію 16 (або 12) слів з V_{32}. На вихід перетворення повертає значення $A' \in V_{384}$ та $B' \in V_{512}$.

У своїй структурі перетворення використовує наступний набір операцій: XOR, побітове ТА, циклічний зсув, додавання за модулем 2^{32}, а також множення на 3 і 5 за цим же модулем. Саме наявністю останніх операцій таке перетворення є цікавим, бо це напряму стосується теми дослідження, що буде проведено у другому розділі.

В результаті дослідження було встановлено, що нехай наразі такий розпізнавач не є повністю застосовним до геш-функції Shabal, проте він обходив доведення захищеності цієї геш-функції.

1.5 Попередні результати аналізу ускладнюючих функцій геш-функції Shabal

У геш-функції Shabal, що була запропонована як учасник конкурсу SHA-3 [10], для побудови псевдовипадкових перестановок були використані ускладнюючі функції $3x \bmod 2^n$ та $5x \bmod 2^n$. Ціллю другого розділу цієї роботи буде побудова покращених обертальних оцінок для цієї функції. Для повноти дослідження, необхідним буде ознайомитись з попередніми, неповними, дослідженнями цих функцій.

Обертальний аналіз функції множення на три

Так, у вже згаданій роботі Гіллеса ван Ассхе [9] було наведено без доведення ймовірність проходження пари обертання через функцію $f_1(x)$ для значень $n = 32$ та $r = 1$. Наведена оцінка виглядала так:

$$rp^{f_1}(1) = \frac{2^{32} - 1}{3 \cdot 2^{32}} \approx \frac{1}{3}.$$

Також, в одній з нещодавніх робіт [11] наведено результати застосування обертального криптоаналізу до функції $f_1(x)$ для довільного значення n та для фіксованих значень обертання $r = 1$ та $r = n - 1$. Обчислені ймовірності мали такий вигляд:

$$\begin{aligned} rp^{f_1}(1) &= \frac{2^n + (-1)^{n+1}}{3 \cdot 2^n}, \\ rp^{f_1}(n-1) &= \frac{2^n + (-1)^{n+1}}{3 \cdot 2^n}. \end{aligned}$$

Обертальний аналіз функції множення на п'ять

Гіллес ван Ассхе також без доведення сформулював [9] оцінку обертальної ймовірності для функції $f_2(x) = 5x \bmod 2^n$ для значення обертання $r = 1$ та довжини векторів $n = 32$:

$$rp^{f_2}(1) = \frac{3 \cdot 2^{32} - 8}{10 \cdot 2^{32}} \approx \frac{3}{10}.$$

Висновки до розділу 1

В ході цього розділу було розглянуто ідею обертального криптоаналізу, розвиток його досліджень та основні твердження, що застосовуються в обертальному криптоаналізі. Було проведено порівняння результатів, отриманих різними авторами на цю тему.

Також були розглянуті приклади застосування даного типу атак на різні криптопримітиви, включно із функцією, що буде розглянута у практичній частині цієї роботи. Надалі, за допомогою інформації, розглянутої в цьому розділі, буде доповнено оцінки ймовірності для функцій вигляду $f_1(x) = 3x \bmod 2^n$ та $f_2(x) = 5x \bmod 2^n$.

2 ЗАСТОСУВАННЯ ОБЕРТАЛЬНОГО КРИПТОАНАЛІЗУ ДО УСКЛАДНЮЮЧИХ ФУНКЦІЇ ГЕШ-ФУНКЦІЇ SHABAL

В межах цього розділу буде сформульовано і доведено оцінку ймовірності проходження пар обертання через функцію $f_1(x) = 3x \bmod 2^n$ для довільного значення n та для значень $r = 2$, $r = 3$ і загального випадку $r \in \{2, \ldots, n-2\}$. Також, буде побудовано оцінки для проходження пар обертання через функцію $f_2(x) = 5x \bmod 2^n$ для довільного значення n та обертань $r = 1$ Та $r = n-1$. Коректність отриманих результатів буде підтверджено за допомогою використання програмного засобу для підрахунку практичної кількості проходжень пар обертання через функції.

2.1 Формулювання допоміжних тверджень

Для доведення основних результатів необхідно сформулювати додаткові твердження про співпадіння бітів переносу з певними бітами вхідного вектора.

Допоміжна лема для функції множення на три

Розглянемо побітове представлення значення $y = 3x \mod 2^n$ для загального вигляду $x \in V_n$, яке наведено у таблиці 2.1.

У цій таблиці c_i – біти переносу операції $x + 2x$, що обчислюються як

$$\forall i \in \{2, \ldots, n-1\} : \; c_i = \left\lfloor \frac{x_{i-1} + x_{i-2} + c_{i-1}}{2} \right\rfloor. \tag{2.1}$$

Таблиця 2.1 – Опис операції множення на три у двійковому представленні.

c	c_{n-1}	c_{n-2}	c_{n-3}	$\dots$	c_{k+1}	c_k	c_{k-1}	$\dots$	c_2	$c_1 = 0$	$c_0 = 0$
x	x_{n-1}	x_{n-2}	x_{n-3}	$\dots$	x_{k+1}	x_k	x_{k-1}	$\dots$	x_2	x_1	x_0
$2x$	x_{n-2}	x_{n-3}	x_{n-4}	$\dots$	x_k	x_{k-1}	x_{k-2}	$\dots$	x_1	x_0	0
$3x$	y_{n-1}	y_{n-2}	y_{n-3}	$\dots$	y_{k+1}	y_k	y_{k-1}	$\dots$	y_2	y_1	y_0

А y_i – біти результату функції $3x$, що обчислюються так:

$$\begin{aligned} y_0 &= x_0; \\ \forall i \in \{1, \dots, n-1\}: \; y_i &= x_i \oplus x_{i-1} \oplus c_i. \end{aligned} \tag{2.2}$$

За своєю природою біти переносу в цій операції формують ланцюг Маркова, тобто значення наступного біту переносу залежить тільки від значення попереднього, а не від усієї послідовності.

Розглянемо подію $x_{k-1} = c_k = 1$ і позначимо ймовірність $p_k = \Pr\{x_{k-1} = c_k = 1\}$. Сформулюємо і доведемо лему про значення цієї імовірності для довільного значення k.

Лема 2.1. *Для довільного $k \in \{2, \dots, n-1\}$ ймовірність p_k можна обчислити таким чином:*

$$p_k = \frac{12\hat{p}_2 + (-2)^k - 4}{3 \cdot (-2)^k}, \tag{2.3}$$

$$\text{де } \hat{p}_2 = \begin{cases} \dfrac{1}{4}, & \text{якщо } k \text{ — парне,} \\ \dfrac{1}{2}, & \text{інакше.} \end{cases}$$

Доведення. Введемо додаткові позначення:

$$p_{i,0} = \Pr\{x_{i-1} = c_i = 0\},$$
$$p_{i,1} = \Pr\{x_{i-1} = c_i = 1\}.$$

Можна побачити, що $p_k = p_{k,1}$.

Для довільного c_i, де $2 < i < k$, розглянемо такі випадки.

1) Якщо $x_{i-1} = 1$, то, за формулою (2.1), $c_i = x_{i-1} = 1$ тоді й тільки тоді, коли $x_{i-2} + c_{i-1} \neq 0$, а отже:

$$p_{i,1} = \Pr\{x_{i-1} = 1\} \cdot \Pr\{x_{i-2} + c_{i-1} \neq 0\} = \frac{1}{2}(1 - p_{i-1,0}).$$

2) Якщо $x_{i-1} = 0$, то, за формулою (2.1), $c_i = x_{i-1} = 0$ тоді й тільки тоді, коли $x_{i-2} + c_{i-1} \neq 1$, а отже:

$$p_{i,0} = \Pr\{x_{i-1} = 0\} \cdot \Pr\{x_{i-2} + c_{i-1} \neq 1\} = \frac{1}{2}(1 - p_{i-1,1}).$$

Виникла рекурента зі змінними початковими значеннями: $p_{2,1}$ та $p_{2,0}$. Іншими словами, якщо для досягнення початкового значення необхідна парна кількість кроків (обчислення рекуренти) — тоді початковим значенням буде $p_{2,1}$, а якщо непарна кількість кроків — тоді $p_{2,0}$. Парність кількості кроків можна обчислити із парності значення k, бо кількість кроків рекуренти становить $k - 2$. Оскільки вигляд рекуренти однаковий і змінюється тільки початкове значення, тоді зведемо ці дві рекуренти до однієї:

$$\hat{p}_i = \frac{1}{2}(1 - \hat{p}_{i-1}). \tag{2.4}$$

Позначимо для певного фіксованого значення k: $\hat{p}_2 = p_{2,1}$, якщо k — парне і $\hat{p}_2 = p_{2,0}$, якщо k — непарне.

З таблиці 2.1 можна побачити, що значення другого біту переносу обчислюється як

$$c_2 \;=\; \lfloor (x_1 + x_0 + 0)/2 \rfloor = x_1 \,\&\, x_0,$$

а отже отримаємо початкові значення, обчисливши відповідні ймовірності.

1) $c_2 = x_1 = 1$ тоді й тільки тоді, коли $x_1 = 1$ і $x_0 = 1$. Ймовірність цієї події дорівнює $1/4$.

2) $c_2 = x_1 = 0$ тоді й тільки тоді, коли $x_1 = 0$ (значенням x_0 можна

знехтувати). Ймовірність цієї події дорівнює $1/2$.

З цього можна зробити висновок, що $\hat{p}_2 = \frac{1}{4}$, якщо k — парне і $\hat{p}_2 = \frac{1}{2}$, якщо k — непарне. Після чого, розписавши рекуренту 2.4 для певного значення k і згорнувши отриману суму за формулою суми геометричної прогресії з часткою $1/2$, ми отримаємо таку формулу:

$$p_k = \hat{p}_k = \frac{12\hat{p}_2 + (-2)^k - 4}{3 \cdot (-2)^k}.$$

□

Допоміжна лема для функції множення на п'ять

Так само як і для попередньої леми, розглянемо побітове представлення $y = 5x \mod 2^n$ для загального вигляду $x \in V_n$, його наведено у таблиці 2.2.

Таблиця 2.2 – Опис операції множення на п'ять у двійковому представленні.

c	c_{n-1}	c_{n-2}	c_{n-3}	$\ldots$	c_{k+1}	c_k	c_{k-1}	$\ldots$	c_3	$c_2 = 0$	$c_1 = 0$	$c_0 = 0$
x	x_{n-1}	x_{n-2}	x_{n-3}	$\ldots$	x_{k+1}	x_k	x_{k-1}	$\ldots$	x_3	x_2	x_1	x_0
$4x$	x_{n-3}	x_{n-4}	x_{n-5}	$\ldots$	x_k	x_{k-1}	x_{k-2}	$\ldots$	x_1	x_0	0	0
$5x$	y_{n-1}	y_{n-2}	y_{n-3}	$\ldots$	y_{k+1}	y_k	y_{k-1}	$\ldots$	y_3	y_2	y_1	y_0

Для подальшого доведення твердження біти переносу краще визначити через функцію мажоризації:

$$\begin{aligned} c_0 = c_1 = c_2 &= 0; \\ c_3 &= x_2 \,\&\, x_0; \\ \forall i \in \{4, \ldots, n-1\}: \; c_i &= \langle x_{i-1}, x_{i-3}, c_{i-1} \rangle. \end{aligned} \tag{2.5}$$

Біти результату y_i функції $5x$, обчислюються так:

$$\begin{aligned} y_0 &= x_0; \\ y_1 &= x_1; \\ y_2 &= x_2 \oplus x_0; \\ \forall i \in \{3, \ldots, n-1\}: \; y_i &= x_i \oplus x_{i-2} \oplus c_i. \end{aligned} \tag{2.6}$$

Розглянемо подію $x_{k-2} = c_k$ і позначимо ймовірність $q_k = \Pr\{x_{k-2} = c_k\}$. Сформулюємо і доведемо лему про значення цієї імовірності для довільного значення k.

Лема 2.2. *Для довільного $k \in \{3, \ldots, n-1\}$ ймовірність q_k можна обчислити таким чином:*

$$q_k = \frac{3 \cdot 2^k + (-1)^{\lfloor \frac{k}{2} \rfloor} \cdot \left(3 + (-1)^{k+1}\right)}{5 \cdot 2^k}. \tag{2.7}$$

Доведення. Розглянемо подію $c_k = x_{k-2}$ для довільного $k \geqslant 5$. Скориставшись визначенням бітів переносу (2.5) розпишемо значення c_k:

$$c_k = \langle x_{k-1}, x_{k-3}, c_{k-1} \rangle = \langle x_{k-1}, x_{k-3}, \langle x_{k-2}, x_{k-4}, c_{k-2} \rangle \rangle.$$

Розпишемо всі можливі варіанти виконання цієї умови та обчислимо їх імовірності.

1) Якщо $x_{k-2} = 0$, а також $x_{k-1} = 0$, то залежно від значення x_{k-3} є два варіанти.

– $x_{k-3} = 0$: в цьому випадку, незалежно від всіх інших значень, $c_k = 0$. Ймовірність цієї події дорівнює:

$$\Pr\{x_{k-2} = 0\} \cdot \Pr\{x_{k-1} = 0\} \cdot \Pr\{x_{k-3} = 0\} = \frac{1}{8}.$$

– $x_{k-3} = 1$: в такому випадку нам необхідно, щоб хоча б одне зі значень x_{k-1} або c_{k-2} дорівнювало 0. Оскільки значення c_{k-2} залежить від

x_{k-3}, для зручності розпишемо ймовірність цього ланцюга таким чином:

$$\Pr\{x_{k-2}=0\}\cdot\Pr\{x_{k-1}=0\}\cdot\Pr\{x_{k-3}=1\}\times$$
$$\times\,(1-\Pr\{c_{k-2}=x_{k-4}=1\mid x_{k-3}=1\})=$$
$$=\frac{1}{8}\cdot(1-\Pr\{c_{k-2}=x_{k-4}=1\mid x_{k-3}=1\}).$$

2) Якщо ж $x_{k-2}=0$, а $x_{k-1}=1$, то $c_{k-2}=0$ тоді й тільки тоді, коли $x_{k-3}=0$, а також хоча б одне зі значень x_{k-1} або c_{k-2} дорівнює 0. Ймовірність цього обчислюється так:

$$\Pr\{x_{k-2}=0\}\cdot\Pr\{x_{k-1}=1\}\cdot\Pr\{x_{k-3}=0\}\times$$
$$\times\,(1-\Pr\{c_{k-2}=x_{k-4}=1\mid x_{k-3}=0\})=$$
$$=\frac{1}{8}\cdot(1-\Pr\{c_{k-2}=x_{k-4}=1\mid x_{k-3}=0\}).$$

3) Для значень $x_{k-2}=1$ та $x_{k-1}=0$ виникає симетричний випадок до попереднього. В такому випадку, необхідними до виконання є умови $x_{k-3}=1$ та що хоча б одне зі значень x_{k-1} або c_{k-2} дорівнює 1:

$$\Pr\{x_{k-2}=1\}\cdot\Pr\{x_{k-1}=0\}\cdot\Pr\{x_{k-3}=1\}\times$$
$$\times\,(1-\Pr\{c_{k-2}=x_{k-4}=0\mid x_{k-3}=1\})=$$
$$=\frac{1}{8}\cdot(1-\Pr\{c_{k-2}=x_{k-4}=0\mid x_{k-3}=1\}).$$

4) І для останнього випадку $x_{k-2}=1$, $x_{k-1}=1$ виникає симетричний випадок до першого пункту.

– $x_{k-3}=1$: незалежно від всіх інших значень, $c_k=1$. Ймовірність цієї події дорівнює:

$$\Pr\{x_{k-2}=1\}\cdot\Pr\{x_{k-1}=1\}\cdot\Pr\{x_{k-3}=1\}=\frac{1}{8}.$$

– $x_{k-3}=0$: необхідно, щоб хоча б одне зі значень x_{k-1} або c_{k-2}

дорівнювало 1.

$$\Pr\{x_{k-2}=1\}\cdot\Pr\{x_{k-1}=1\}\cdot\Pr\{x_{k-3}=0\}\times$$
$$\times\left(1-\Pr\{c_{k-2}=x_{k-4}=0\mid x_{k-3}=0\}\right)=$$
$$=\frac{1}{8}\cdot\left(1-\Pr\{c_{k-2}=x_{k-4}=0\mid x_{k-3}=0\}\right).$$

Описані вище умови повністю покривають виконання $c_k = x_{k-2}$ і при цьому є попарно незалежними. Тож, ймовірність цієї події можна обчислити шляхом додавання ймовірностей виконання всіх цих умов. Після низки перетворень із застосуванням формули повної ймовірності та об'єднання умов $c_{k-2} = x_{k-4} = 0$ та $c_{k-2} = x_{k-4} = 1$ в $c_{k-2} = x_{k-4}$ ми отримаємо рекуренту:

$$q_k = \frac{3}{4} - \frac{1}{4}\Pr\{c_{k-2} = x_{k-4}\} = \frac{3}{4} - \frac{1}{4}q_{k-2}. \tag{2.8}$$

Початкові Значення рекуренти визначимо для парного та непарного значення k.

1) $q_3 = \Pr\{c_3 = x_1\}$. Ймовірність цієї події дорівнює ймовірності виконання $x_0 \,\&\, x_2 = x_1$, тобто $\frac{1}{2}$.

2) $q_4 = \Pr\{c_4 = x_2\}$. Розглянувши $c_4 = \langle x_3, x_1, \langle x_2, x_1, 0\rangle\rangle$ та перебравши різні значення бітів, можна вставити, що ймовірність цієї події становить $\frac{5}{8}$.

Звівши рекуренту (2.8) із заданими початковими значеннями до аналітичного вигляду, отримаємо таку формулу:

$$q_k = \frac{(2-4i)\cdot(-i)^k + (2-4i)\cdot i^k + 6\cdot 2^k}{10\cdot 2^k}.$$

Залежно від залишку від ділення k на 4, отримаємо чотири простіші

формули:

$$
\begin{aligned}
k \bmod 4 = 0 \ &: \quad q_k = \frac{3 \cdot 2^k + 2}{5 \cdot 2^k}; \\
k \bmod 4 = 1 \ &: \quad q_k = \frac{3 \cdot 2^k + 4}{5 \cdot 2^k}; \\
k \bmod 4 = 2 \ &: \quad q_k = \frac{3 \cdot 2^k - 2}{5 \cdot 2^k}; \\
k \bmod 4 = 3 \ &: \quad q_k = \frac{3 \cdot 2^k - 4}{5 \cdot 2^k}.
\end{aligned}
$$

Об'єднаємо ці формули і отримаємо кінцевий результат:

$$
q_k = \frac{3 \cdot 2^k + (-1)^{\lfloor k/2 \rfloor} \cdot (3 + (-1)^{k+1})}{5 \cdot 2^k}.
$$

□

2.2 Обертальний криптоаналіз функції множення на три для окремих значень обертання

Для розуміння природи проходження пар обертання через функцію $f_1(x) = 3x \bmod 2^n$, розглянемо частковий випадок для обертання $r = 2$. Результатом аналізу є таке твердження.

Твердження 2.1. *Ймовірність проходження пари обертання через функцію* $f_1(x) = 3x \mod 2^n$ *при обертанні* $r = 2$ *дорівнює*

$$
rp^{f_1}(2) = \frac{1}{6} + \frac{2^{1+(n \bmod 2)}}{3 \cdot 2^n}.
$$

Доведення.

Розглянемо таблиці обчислення значень $y = 3x \bmod 2^n$, $y \lll 2$ та $l = 3(x \lll 2) \bmod 2^n$.

Таблиця 2.3 – Двійкове представлення $y = 3x \bmod 2^n$ та $y \lll 2$.

c	c_{n-1}	c_{n-2}	c_{n-3}	$\dots$	c_2	$c_1 = 0$	$c_0 = 0$
x	x_{n-1}	x_{n-2}	x_{n-3}	$\dots$	x_2	x_1	x_0
$2x$	x_{n-2}	x_{n-3}	x_{n-4}	$\dots$	x_1	x_0	0
$3x$	y_{n-1}	y_{n-2}	y_{n-3}	$\dots$	y_2	y_1	y_0
$3x \lll 2$	y_{n-3}	y_{n-4}	y_{n-3}	$\dots$	y_0	y_{n-1}	y_{n-2}

Таблиця 2.4 – Двійкове представлення $l = 3(x \lll 2) \bmod 2^n$.

c'	c'_{n-1}	c'_{n-2}	$\dots$	c'_3	c'_2	$c'_1 = 0$	$c'_0 = 0$
$x \lll 2$	x_{n-3}	x_{n-4}	$\dots$	x_1	x_0	x_{n-1}	x_{n-2}
$2(x \lll 2)$	x_{n-4}	x_{n-5}	$\dots$	x_0	x_{n-1}	x_{n-2}	0
$3(x \lll 2)$	l_{n-1}	l_{n-2}	$\dots$	l_3	l_2	l_1	l_0

Зіставивши відповідні біти векторів $3x \lll 2$ (таблиця 2.3) та $3(x \lll 2)$ (таблиця 2.4) отримаємо такі умови:

$$\begin{aligned} y_{n-2} &= l_0; \\ y_{n-1} &= l_1; \\ y_0 &= l_2; \\ y_1 &= l_3; \\ \forall i, 1 < i < n-2: \quad y_i &= l_{i+2}. \end{aligned}$$

Розпишемо значення бітів за формулою (2.2) та скоротимо відповідні значення:

$$x_{n-2} \oplus x_{n-3} \oplus c_{n-2} = x_{n-2} \Rightarrow \quad x_{n-3} \oplus c_{n-2} = 0; \tag{2.9}$$

$$x_{n-1} \oplus x_{n-2} \oplus c_{n-1} = x_{n-1} \oplus x_{n-2} \Rightarrow \quad c_{n-1} = 0; \tag{2.10}$$

$$x_0 = x_0 \oplus x_{n-1} \oplus c'_2 \Rightarrow \quad x_{n-1} \oplus c'_2 = 0; \tag{2.11}$$

$$x_1 \oplus x_0 = x_1 \oplus x_0 \oplus c'_3 \Rightarrow \quad c'_3 = 0; \tag{2.12}$$

$$x_i \oplus x_{i-1} \oplus c_i = x_i \oplus x_{i-1} \oplus c'_{i+2} \Rightarrow \quad c_i = c'_{i+2}, \text{ для } 1 < i < n-2. \tag{2.13}$$

Отримано 5 умов, необхідних для проходження обертальної пари

через систему. Спростимо їх та розіб'ємо на попарно незалежні між собою умови.

1) Якщо об'єднати умови (2.9) та (2.10), то можна отримати умови дещо спрощеного вигляду:

$$c_{n-1} = 0 \text{ та } x_{n-3} = c_{n-2} \Leftrightarrow \left|\text{З формули (2.1)}\right| \Leftrightarrow x_{n-3} = 0 \text{ та } c_{n-2} = 0.$$

2) Аналогічно можна перетворити умови (2.11) та (2.12):

$$c'_3 = 0 \text{ та } x_{n-1} = c'_2 \Leftrightarrow \left|\text{За формулою (2.1)}\right| \Leftrightarrow x_{n-1} = 0 \text{ та } c'_2 = 0.$$

3) Тепер можна показати, що умова (2.13) виконується автоматично при $x_{n-1} = 0$. Розглянемо цю умову для $i = 2$:

$$c'_4 = \lfloor (x_1 + x_0 + (x_0 \,\&\, x_{n-1}))/2 \rfloor = \left| x_{n-1} = 0 \right| = \lfloor (x_1 + x_0 + 0)/2 \rfloor = c_2.$$

З виконання $c_2 = c'_4$ випливає те, що всі наступні біти переносу почнуть збігатися, оскільки при їх обчисленні будуть використовуватись однакові біти вектора x. Отже, ми можемо позбутися умови (2.13).

4) Розглянемо значення $c'_2 = \lfloor (x_{n-1} + x_{n-2} + c'_1)/2 \rfloor \rfloor$. З таблиці 2.4 можна побачити, що $c'_1 = 0$. Тоді, з умови $x_{n-1} = 0$ випливатиме $c'_2 = 0$. Таким чином, умову $c'_2 = 0$ також можна відкинути.

5) Останніми залежними між собою умовами залишаються $c_{n-2} = 0$ та $x_{n-3} = 0$. Позбудемося цієї залежності, знайшовши необхідну до виконання умову, за якої $c_{n-2} = 0$ при $x_{n-3} = 0$:

$$c_{n-2} = 0 \Leftrightarrow \left|\text{З формули (2.1)}\right| \Leftrightarrow \lfloor (x_{n-3} + x_{n-4} + c_{n-3}) = 0 \rfloor \Leftrightarrow$$

$$\Leftrightarrow \left| x_{n-3} = 0 \right| \Leftrightarrow x_{n-4} = 0 \text{ або } c_{n-3} = 0 \Leftrightarrow \neg(x_{n-4} = c_{n-3} = 1).$$

В результаті цих перетворень ми отримали три незалежні між собою

необхідні і достатні умови проходження пари обертання через функцію $f_1(x)$ при обертанні $r = 2$: $x_{n-1} = 0$, $x_{n-3} = 0$ та $\neg(x_{n-4} = c_{n-3} = 1)$. Тож, ймовірність можна обчислити таким чином:

$$rp^{f_1}(2) = \Pr\{x_{n-1} = 0\} \cdot \Pr\{x_{n-3} = 0\} \cdot \Pr\{\neg(x_{n-4} = c_{n-3} = 1)\} =$$
$$= \frac{1}{2} \cdot \frac{1}{2} \cdot (1 - \Pr\{x_{n-4} = c_{n-3} = 1\}) = \left|\text{Лема 2.1}\right| =$$
$$= \frac{1}{4} \cdot \left(1 - \frac{12\hat{p}_2 + (-2)^{n-3} - 4}{3 \cdot (-2)^{n-3}}\right).$$

Після низки арифметичних перетворень, отримаємо наступний результат обертальної ймовірності:

$$rp^{f_1}(2) = \frac{1}{6} + \frac{2^{1+(n \bmod 2)}}{3 \cdot 2^n}.$$

□

Користуючись аналогічним підходом отримаємо оцінку для обертання на три.

Твердження 2.2. *Ймовірність проходження пари обертання через функцію* $f_1(x) = 3x \mod 2^n$ *при обертанні* $r = 3$ *дорівнює*

$$rp^{f_1}(3) = \frac{1}{8} + \frac{2^{1-(n \bmod 2)}}{2^n}.$$

Сформульовані твердження пізніше будуть використані для перевірки та підтвердження оцінки для довільного обертання.

2.3 Обертальний криптоаналіз функції множення на три в загальному випадку

Узагальнюючи судження, що були застосовані в доведенні для значень обертання $r = 2$ (твердження 2.1) та $r = 3$ (твердження 2.2), можна сформулювати теорему про вигляд формули ймовірності для

загального випадку для значень обертання $r \in \{2, \ldots, n-2\}$.

Теорема 2.1. *Для довільного значення n, ймовірність проходження пари обертання через функцію $f_1(x) = 3x \bmod 2^n$ при довільному значенні обертання $r \in \{2, \ldots, n-2\}$ можна обчислити за такою формулою:*

$$rp^{f_1}(r) = \frac{1}{9}\left(1 + \frac{2^{1-(n-r) \bmod 2}}{2^{n-r}}\right)\left(1 + \frac{2^{1-r \bmod 2}}{2^r}\right).$$

Доведення.

Позначимо $y = 3x \bmod 2^n$ і $l = 3(x \lll r) \bmod 2^n$.

Таблиця 2.5 – Двійкове представлення $y = 3x \bmod 2^n$ та $y \lll r$.

c	c_{n-1}	c_{n-2}	c_{n-3}	$\ldots$	c_{r+1}	c_r	c_{r-1}	$\ldots$	c_2	$c_1 = 0$	$c_0 = 0$
x	x_{n-1}	x_{n-2}	x_{n-3}	$\ldots$	x_{r+1}	x_r	x_{r-1}	$\ldots$	x_2	x_1	x_0
$2x$	x_{n-2}	x_{n-3}	x_{n-4}	$\ldots$	x_r	x_{r-1}	x_{r-2}	$\ldots$	x_1	x_0	0
$3x$	y_{n-1}	y_{n-2}	y_{n-3}	$\ldots$	y_{r+1}	y_r	y_{r-1}	$\ldots$	y_2	y_1	y_0
$3x \lll r$	y_{n-1-r}	y_{n-2-r}	y_{n-3-r}	$\ldots$	y_1	y_0	y_{n-1}	$\ldots$	y_{n-r+2}	y_{n-r+1}	y_{n-r}

Таблиця 2.6 – Двійкове представлення $l = 3(x \lll r) \bmod 2^n$.

c'	c'_{n-1}	c'_{n-2}	c'_{n-3}	$\ldots$	c'_{r+1}	c'_r	c'_{r-1}	$\ldots$	c'_2	$c'_1 = 0$	$c'_0 = 0$
$x \lll r$	x_{n-1-r}	x_{n-2-r}	x_{n-3-r}	$\ldots$	x_1	x_0	x_{n-1}	$\ldots$	x_{n-r+2}	x_{n-r+1}	x_{n-r}
$2(x \lll r)$	x_{n-2-r}	x_{n-3-r}	x_{n-4-r}	$\ldots$	x_1	x_0	x_{n-1}	$\ldots$	x_{n-r+1}	x_{n-r}	0
$3(x \lll r)$	l_{n-1}	l_{n-2}	l_{n-3}	$\ldots$	l_{r+1}	l_r	l_{r-1}	$\ldots$	l_2	l_1	l_0

Розглянемо таблиці 2.5 та 2.6, які є побітовим представленням застосування функції $f_1(x) = 3x \bmod 2^n$ до вектора x зі зсувом і після цього застосуванням $f_1(x)$ до зсунутого вектора $x \lll r$. Щоб порахувати ймовірність події $y \lll r = l$, необхідно зіставити ці два вектори побітово.

З таблиці 2.5 можна отримати такі співвідношення:

$$\begin{aligned}
&y_0 = x_0, && c_0 = c_1 = 0;\\
&\forall i,\ 1 < i < n: && y_i = x_i \oplus x_{i-1} \oplus c_i;\\
&\forall i,\ 1 < i < n-1: && c_{i+1} = \left\lfloor \frac{x_i + x_{i-1} + c_i}{2} \right\rfloor.
\end{aligned}$$

З таблиці 2.6 можна отримати такі співвідношення:

$$\begin{aligned}
&l_0 = x_{n-r}, && c'_0 = c'_1 = 0;\\
&l_1 = x_{n-r+1} \oplus x_{n-r}, && c'_2 = x_{n-r+1} \,\&\, x_{n-r};\\
&\forall i,\ 1 < i < r: && l_i = x_{n-r+i} \oplus x_{n-r+i-1} \oplus c'_i,\\
&\forall i,\ 1 < i < r-1: && c'_{i+1} = \left\lfloor \frac{x_{n-r+i} + x_{n-r+i-1} + c'_i}{2} \right\rfloor;\\
&l_r = x_0 \oplus x_{n-1} \oplus c'_r, && c'_{r+1} = \left\lfloor \frac{x_0 + x_{n-1} + c'_r}{2} \right\rfloor;\\
&\forall i,\ r < i < n: && l_i = x_{i-r} \oplus x_{i-r-1} \oplus c'_i,\\
&\forall i,\ r < i < n-1: && c'_{i+1} = \left\lfloor \frac{x_{i-r} + x_{i-r-1} + c'_i}{2} \right\rfloor;
\end{aligned}$$

Зіставивши відповідні біти векторів $y \lll r$ та l, отримаємо такі умови:

$$\begin{aligned}
l_0 = y_{n-r}, \qquad l_1 &= y_{n-r+1};\\
\forall i,\ 1 < i < r: \qquad l_i &= y_{n-r+i};\\
l_r = y_0, \qquad l_{r+1} &= y_1;\\
\forall i,\ r+1 < i < n: \qquad l_i &= y_{i-r}.
\end{aligned}$$

Розписавши їх значення відповідно до зазначеного вище, отримаємо умови, необхідні для виконання рівності:

$$x_{n-r-1} = c_{n-r}; \tag{2.14}$$

$$c_{n-r+1} = 0; \tag{2.15}$$

$$x_{n-1} = c'_r; \tag{2.16}$$

$$c'_{r+1} = 0; \tag{2.17}$$

$$\forall i,\, 1 < i < r: \quad c'_i = c_{n-r+i}; \tag{2.18}$$

$$\forall i,\, r+1 < i < n: \quad c'_i = c_{i-r}. \tag{2.19}$$

Зменшимо кількість цих умов, встановивши між ними зв'язок.

1) З умови (2.15) випливає умова (2.18).

Розглянемо для $i = 2$:

$$c_{n-r+2} = \lfloor (x_{n-r+1} + x_{n-r} + c_{n-r+1})/2 \rfloor =$$

$$= \left| \text{за ум. (2.15)} \right| = \lfloor (x_{n-r+1} + x_{n-r})/2 \rfloor =$$

$$= x_{n-r+1} \,\&\, x_{n-r} = c'_2.$$

Для всіх інших значень i з проміжку $\overline{3 \ldots r-1}$ умова виконуватиметься автоматично, оскільки починаючи з $i = 2$ значення c'_i та c_{n-r+i} почнуть збігатися завдяки показаному вище.

2) Аналогічним чином можна показати, що з умови (2.17) випливає умова (2.19).

Для значення $i = r + 2$:

$$c'_{r+2} = \lfloor (x_{r+2-r+1} + x_{r+2-r-2} + c_{r+1})/2 \rfloor =$$

$$= \left| \text{за ум. (2.17)} \right| = \lfloor (x_1 + x_0)/2 \rfloor =$$

$$= x_1 \,\&\, x_0 = c_2.$$

А отже, для всіх значень i з проміжку $\overline{r+3 \ldots n-1}$ ця умова також виконується, оскільки значення бітів переносу починають збігатися через

однакове значення відповідних бітів вектора x.

3) Поєднаємо умови (2.14) та (2.15):

$$x_{n-r-1} = c_{n-r} \,\&\, c_{n-r+1} = 0 \Leftrightarrow$$
$$\Leftrightarrow \left| c_{n-r+1} = \lfloor (x_{n-r} + x_{n-r-1} + c_{n-r})/2 \rfloor \right| \Leftrightarrow$$
$$\Leftrightarrow x_{n-r-1} = 0 \text{ та } c_{n-r} = 0. \tag{2.20}$$

4) Поєднаємо умови (2.16) та (2.17):

$$x_{n-1} = c'_r \,\&\, c'_{r+1} = 0 \Leftrightarrow$$
$$\Leftrightarrow \left| c'_{r+1} = \lfloor (x_0 + x_{n-1} + c'_r)/2 \rfloor \right| \Leftrightarrow$$
$$\Leftrightarrow x_{n-1} = 0 \text{ та } c_r = 0. \tag{2.21}$$

Позбудемося залежності між отриманими умовами 2.20 та 2.21, об'єднавши їх.

– З вигляду формули (2.1) для обчислення бітів переносу, умова $c_{n-r} = 0$, при виконанні $x_{n-r-1} = 0$, є рівносильною $x_{n-r-2} \,\&\, c_{n-r-1} = 0$, що за визначенням операції ТА є еквівалентним $\neg(x_{n-r-2} = c_{n-r-1} = 1)$.

– Аналогічно умова $c'_r = 0$, при виконанні $x_{n-1} = 0$, є рівносильною $x_{n-2} \,\&\, c'_{r-1} = 0$. А ця подія є еквівалентною до $\neg(x_{n-2} = c'_{r-1} = 1)$.

Таким чином, ми отримали чотири незалежні між собою умови, обчислимо їх імовірності поодинці.

1) $x_{n-1} = 0$: за припущенням про рівноймовірний розподіл векторів x, імовірність цієї події становить $\Pr\{x_{n-1} = 0\} = \frac{1}{2}$.

2) $x_{n-r-1} = 0$: аналогічним чином імовірність цієї події становить $\Pr\{x_{n-r-1} = 0\} = \frac{1}{2}$.

3) Ймовірність події $\neg(x_{n-r-2} = c_{n-r-1} = 1)$ можна обчислити за

допомогою леми 2.1:

$$\Pr\{x_{n-r-2}\,\&\,c_{n-r-1}=0\}=$$
$$=1-\Pr\{x_{n-r-2}=c_{n-r-1}=1\}=1-p_{n-r-1}=$$
$$=1-\frac{12\hat{p}_2+(-2)^{n-r-1}-4}{3\cdot(-2^{n-r-1})},$$

де $\hat{p}_2=\frac{1}{4}$, якщо $(n-r-1)$ — парне, і $\hat{p}_2=\frac{1}{2}$, якщо це значення непарне. Після низки арифметичних перетворень отримаємо такий результат:

$$\Pr\{x_{n-r-2}\,\&\,c_{n-r-1}=0\}=\frac{2}{3}\left(1+\frac{2^{1-(n-r)\bmod 2}}{2^{n-r}}\right).$$

4) Для умови $\neg(x_{n-2}=c'_{r-1}=1)$ також застосовна лема 2.1, але вже до зсунутого вектора:

$$\Pr\{x_{n-2}\,\&\,c'_{r-1}=0\}=$$
$$=1-\Pr\{x_{n-2}=c'_{r-1}=1\}=1-p_{r-1}=$$
$$=1-\frac{12\hat{p}_2+(-2)^{r-1}-4}{3\cdot(-2^{r-1})},$$

де $\hat{p}_2=\frac{1}{4}$, якщо $(r-1)$ — парне, і $\hat{p}_2=\frac{1}{2}$, якщо це значення непарне. Провівши аналогічні перетворення, як і в попередньому пункті, отримаємо:

$$\Pr\{x_{n-2}\,\&\,c'_{r-1}=0\}=\frac{2}{3}\left(1+\frac{2^{1-r\bmod 2}}{2^{r}}\right).$$

Враховуючи, що наведені вище події є незалежними й тільки за їх виконання є істинним твердження $y\lll r=l$, то можна стверджувати, що:

$$rp^{f_1}(r) = \Pr\{y \lll r = l\} =$$
$$= \Pr\{x_{n-1} = 0,\, x_{n-r-1} = 0,\, x_{n-r-2} \,\&\, c_{n-r-1} = 0,\, x_{n-2} \,\&\, c'_{r-1} = 0\} =$$
$$= \Pr\{x_{n-1} = 0\} \cdot \Pr\{x_{n-r-1} = 0\} \times$$
$$\times \Pr\{x_{n-r-2} \,\&\, c_{n-r-1} = 0\} \cdot \Pr\{x_{n-2} \,\&\, c'_{r-1} = 0\} =$$
$$= \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{2}{3} \left(1 + \frac{2^{1-(n-r) \bmod 2}}{2^{n-r}}\right) \cdot \frac{2}{3} \left(1 + \frac{2^{1-r \bmod 2}}{2^r}\right) =$$
$$= \frac{1}{9} \left(1 + \frac{2^{1-(n-r) \bmod 2}}{2^{n-r}}\right) \left(1 + \frac{2^{1-r \bmod 2}}{2^r}\right).$$

□

Розглянемо оцінку, отриману в теоремі 2.1. Зафіксувати значення обертання $r = 2$ або $r = 3$, ми отримаємо оцінки ймовірностей, що попередньо були отримані у твердженнях 2.1 та 2.2 відповідно. Іншими словами, отримана теорема є узагальненням тверджень, отриманих в попередньому підрозділі.

Проаналізувавши результат, можна стверджувати, що при фіксованому значенні обертання r на асимптотиці прямує до певної константи.

Твердження 2.3. *Якщо значення обертання набуває значення $r \geqslant 2$, тоді при збільшенні значення довжини двійкових векторів n виконується така збіжність:*

$$rp^{f_1}(r) \to \frac{1}{9} \left(1 + \frac{2^{1-r \bmod 2}}{2^r}\right), \text{ при } n \to \infty.$$

Доведення. Твердження виконується, оскільки другий множник у формулі, зазначеній у теоремі 2.1 при збільшенні значення n починає дуже швидко прямувати до 1, тому ми можемо відкинути цей множник на асимптотиці. □

Підтвердження результату експериментальним шляхом

Скориставшись програмою А.1, обчислимо практичну кількість проходжень пар обертання з обраними обертаннями r через функцію $f(x) = 3x \bmod 2^n$ для довжин векторів $n \in \{20, \ldots, 25\}$, а також для $n = 32$ з обертаннями $r \in \{2, \ldots, 6\}$. Теоретичне значення цієї статистики обчислимо з теореми 2.1 шляхом множення ймовірності на кількість векторів 2^n. Отримані результати порівняємо у таблицях 2.7 та 2.8 відповідно.

Таблиця 2.7 – Порівняння практичної та теоретичної, отриманої за допомогою теореми 2.1, кількості проходження пар обертання через функцію $f_1(x)$ для значень вектора $n \in \{20, \ldots, 25\}$ та $r \in \{7, \ldots, 13\}$.

n	Спосіб обчислення	Значення обертання r						
		7	8	9	10	11	12	13
20	Практ.	117433	117476	116793	116964	116793	117476	117433
	Теор.	117433	117476	116793	116964	116793	117476	117433
21	Практ.	234866	234866	233586	233586	233586	233586	234866
	Теор.	234866	234866	233586	233586	233586	233586	234866
22	Практ.	469689	469732	467001	467172	466489	467172	467001
	Теор.	469689	469732	467001	467172	466489	467172	467001
23	Практ.	939378	939378	934002	934002	932978	932978	934002
	Теор.	939378	939378	934002	934002	932978	932978	934002
24	Практ.	1878713	1878756	1867833	1868004	1865273	1865956	1865273
	Теор.	1878713	1878756	1867833	1868004	1865273	1865956	1865273
25	Практ.	3757426	3757426	3735666	3735666	3730546	3730546	3730546
	Теор.	3757426	3757426	3735666	3735666	3730546	3730546	3730546

Як можна побачити, практичні результати підрахунку кількості пар обертання, що проходять через функцію $f(x) = 3x \bmod 2^n$, збігається з теоретичною оцінкою. Це певною мірою підтверджує коректність побудованої оцінки.

Таблиця 2.8 – Порівняння практичної та теоретичної, отриманої за допомогою теореми 2.1, кількості проходження пар обертання через функцію $f_1(x)$ для довжини вектора $n = 32$ та обертань $r \in \{2, \ldots, 6\}$.

n	Спосіб обчислення	Значення обертання r				
		2	3	4	5	6
32	Практ.	715827884	536870913	536870916	492131673	492131684
	Теор.	715827884	536870913	536870916	492131673	492131684

Також, цікавим є спостереження симетричності результатів, яке дуже помітно у таблиці 2.7. Це явище пов'язане з тим, що побудована оцінка також є коректною, якщо розглядати обертання праворуч, замість обертання ліворуч.

2.4 Обертальний криптоаналіз функції множення на п'ять для окремих значень обертання

Використовуючи схожий до попередніх доведень підхід аналізу властивостей бітових векторів, для яких виконується проходження пар обертання через функції такого вигляду, можна сформулювати та довести оцінку для ймовірності їх проходження через функцію $5x \bmod 2^n$ при фіксованому значенні обертання $r = 1$.

Теорема 2.2. *Ймовірність проходження пари обертання через функцію* $f_2(x) = 5x \bmod 2^n$ *при обертанні* $r = 1$ *можна обчислити за такою формулою:*

$$rp^{f_2}(1) = \frac{3 \cdot 2^{n-1} + (-1)^{\lfloor (n-1)/2 \rfloor} \cdot (3 + (-1)^n)}{5 \cdot 2^n}.$$

Доведення. Аналогічно до попередніх доведень, розглянемо таблиці побітового обчислення значень $y = 5x \bmod 2^n$, $y \lll 1$ та $l = 5(x \lll 1) \bmod 2^n$.

Таблиця 2.9 – Двійкове представлення $y = 5x \bmod 2^n$ та $y \lll 1$.

c	c_{n-1}	c_{n-2}	c_{n-3}	$\dots$	c_3	$c_2 = 0$	$c_1 = 0$	$c_0 = 0$
x	x_{n-1}	x_{n-2}	x_{n-3}	$\dots$	x_3	x_2	x_1	x_0
$4x$	x_{n-3}	x_{n-4}	x_{n-5}	$\dots$	x_1	x_0	0	0
$5x$	y_{n-1}	y_{n-2}	y_{n-3}	$\dots$	y_3	y_2	y_1	y_0
$5x \lll 2$	y_{n-2}	y_{n-3}	y_{n-4}	$\dots$	y_2	y_1	y_0	y_{n-1}

Таблиця 2.10 – Двійкове представлення $l = 5(x \lll 1) \bmod 2^n$.

c'	c'_{n-1}	c'_{n-2}	c'_{n-3}	$\dots$	c'_3	$c'_2 = 0$	$c'_1 = 0$	$c'_0 = 0$
$x \lll 1$	x_{n-2}	x_{n-3}	x_{n-4}	$\dots$	x_2	x_1	x_0	x_{n-1}
$4(x \lll 1)$	x_{n-4}	x_{n-5}	x_{n-6}	$\dots$	x_0	x_{n-1}	0	0
$5(x \lll 1)$	l_{n-1}	l_{n-2}	c'_{n-2}	$\dots$	l_3	l_2	l_1	l_0

Співставивши відповідні біти з таблиць 2.9 та 2.10 отримаємо такі умови:

$$\begin{aligned} y_{n-1} &= l_0; \\ y_0 &= l_1; \\ y_1 &= l_2; \\ y_2 &= l_3; \\ \forall i, 2 < i < n-1 \ : \ y_i &= l_{i+1}. \end{aligned}$$

Розписавши ліві і праві сторони рівностей за формулою (2.6) та скоротивши відповідні біти отримаємо спрощені умови:

$$x_{n-3} = c_{n-1}; \tag{2.22}$$

$$x_0 = x_0; \tag{2.23}$$

$$x_{n-1} = 0; \tag{2.24}$$

$$x_1 \,\&\, x_{n-1} = 0; \tag{2.25}$$

$$\forall i, 2 < i < n-1 \ : \ c_i = c'_{i+1}. \tag{2.26}$$

Очевидним є те, що умова (2.23) точно виконується, а з умови (2.24) випливає (2.25). Рівність (2.26) також виконується, що можна показати аналогічно до того, як виконання подібних рівностей було показано в попередніх доведеннях.

Тож, пара обертання проходитиме через функцію f_2 за виконання двох незалежних умов $x_{n-1} = 0$ та $x_{n-3} = c_{n-1}$. Ймовірність виконання другої можна обчислити за допомогою леми 2.2.

$$rp^{f_2}(1) = \Pr\{x_{n-1} = 0\} \cdot \Pr\{x_{n-3} = c_{n-1}\} = \frac{1}{2} q_{n-1} = \\ = \frac{1}{2} \cdot \frac{3 \cdot 2^{n-1} + (-1)^{(n-1)/2} \cdot (3 + (-1)^n)}{5 \cdot 2^{n-1}} = \\ = \frac{3 \cdot 2^{n-1} + (-1)^{(n-1)/2} \cdot (3 + (-1)^n)}{5 \cdot 2^n}.$$

□

Застосовуючи такий же підхід до аналізу можна сформулювати оцінку і для значення обертання $r = n - 1$.

Теорема 2.3. *Ймовірність проходження пари обертання через функцію $f_2(x) = 5x \bmod 2^n$ при обертанні $r = n - 1$ можна обчислити за такою формулою:*

$$rp^{f_2}(n-1) = \frac{3 \cdot 2^{n-1} + (-1)^{\lfloor (n-1)/2 \rfloor} \cdot (3 + (-1)^n)}{5 \cdot 2^n}.$$

Так само як і для $f_1(x)$, можна сформулювати асимптотичну оцінку значення ймовірності проходження пар обертання.

Твердження 2.4. *Зі збільшенням значення n, виконуються такі збіжності:*

$$rp^{f_2}(1) \ \to \ \frac{3}{10}, \ \textit{при } n \to \infty; \\ rp^{f_2}(n-1) \ \to \ \frac{3}{10}, \ \textit{при } n \to \infty.$$

Доведення.

Розпишемо значення $rp^{f_2}(1)$ на суму двох дробів:

$$rp^{f_2}(1) = \frac{3 \cdot 2^{n-1} + (-1)^{\lfloor (n-1)/2 \rfloor} \cdot (3 + (-1)^n)}{5 \cdot 2^n} = \\ = \frac{3}{10} + \frac{(-1)^{\lfloor (n-1)/2 \rfloor} \cdot (3 + (-1)^n)}{5 \cdot 2^n}.$$

Чисельник другого доданку не збільшується зі збільшенням значення n, а знаменник експоненційно збільшується. З цього можна зробити висновок, що другий доданок стрімко прямує до нуля, а отже саме значення імовірності дуже швидко прямує до значення $\frac{3}{10}$.

Аналогічні міркування застосовні і до значення $rp^{f_2}(n-1)$. □

Підтвердження результату експериментальним шляхом

За допомогою програми А.1, так само як і для функції $f_1(x)$, можемо перевірити оцінки, отримані у теоремах 2.2 та 2.3. Оскільки при побудові оцінки було використано розбиття на класи за значенням $k \bmod 4$, буде доречним розглянути значення для чотирьох послідовних значень n: 21, 22, 23 та 24.

Таблиця 2.11 – Порівняння практичних результатів з результатами, отриманими за допомогою теорем 2.2 та 2.3.

n		21		22		23		24	
Спосіб обчислення		Практ.	Теор.	Практ.	Теор.	Практ.	Теор.	Практ.	Теор.
r	1	629146	629146	1258292	1258292	2516582	2516582	5033164	5033164
	$n-1$	629146	629146	1258292	1258292	2516582	2516582	5033164	5033164

Як можна побачити в таблиці, значення обчислені за допомогою сформульованих теорем збігаються зі значеннями, отриманими шляхом перебору. Це підтверджує коректність і демонструє застосовність цих оцінок на практиці.

Висновки до розділу 2

В ході цього розділу побудовано оцінки для ймовірностей проходження обертальних пар через функцію $3x \bmod 2^n$ для фіксованих значень обертання $r = 2$ та $r = 3$. Після цього, користуючись особливостями доведення, поміченими під час побудови оцінки для фіксованих значень обертання, побудовано оцінку для довільного значення обертання у проміжку $2 \leqslant r \leqslant n - 2$. Також, було побудовано часткову оцінку ймовірностей для функції $5x \bmod 2^n$ для випадків обертання на 1 та $n - 1$ бітів. Для доведення основних тверджень і теорем, було сформульовано і доведено допоміжні леми, що пізніше можуть бути використані для доведень в подальшому.

Коректність побудованих оцінок була перевірена за допомогою програми А.1, шляхом співставлення реальної кількості пар обертання, що пройшли через систему, та теоретичної, отриманої за допомогою теорем 2.2 та 2.3.

ВИСНОВКИ

У ході даної роботи було розглянуто основні досягнення аналізу ускладнюючих функцій геш-функції Shabal, оцінки для яких було покращено.

Було одержано аналітичний вираз для обчислення ймовірності проходження пар обертання через функцію множення на три за модулем 2^n для довільної довжини та значень обертання від 2 до $n-2$. Було також встановлено, що, при збільшенні довжини вхідних векторів, отримана ймовірність стрімко прямує до певного фіксованого значення, що залежить від обертання. Так, наприклад, при обертанні $r = 2$, ймовірність приблизно можна обчислювати як $\frac{1}{6}$, а для обертання $r = 3$ це значення становить $\frac{1}{8}$. Такі наближення спрощують аналіз функції і є застосовними у більшості випадків, бо відхилення часто є несуттєвими.

Для функції множення на п'ять, своєю чергою, було одержано та доведено аналітичні формули для обчислення обертальних ймовірностей для векторів довільної довжини та значень обертання $r = 1$ та $r = n - 1$. Так само як і для випадку множення на три, значення отриманих імовірностей при достатньо великому значенні n можна наблизити до сталих. Для обох випадків значення ймовірності прямує до $\frac{3}{10}$.

Для доведення основних тверджень та теорем, також було сформульовано та доведено окремі допоміжні леми. Всі одержані результати було перевірено на коректність за допомогою програмного застосунку, розробленого на мові програмування C++, шляхом порівняння теоретичних результатів, отриманих за допомогою сформульованих теорем, та практичних, отриманих шляхом перебору.

Надалі, результати дослідження, а особливо допоміжні леми, можна застосувати для покращення аналізу функції множення на п'ять, а також для побудови оцінки для довільної функції вигляду $(2^s + 1)x \bmod 2^n$.

ПЕРЕЛІК ПОСИЛАНЬ

[1] Бондар П.О. «Обертальний криптоаналіз однієї з ускладнюючих функцій геш-функції Shabal». В: *XXII Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» (Україна, м. Київ, 17 травня 2024 р.) : матеріали конференції*. Київ: КПІ ім. Ігоря Сікорського, Видавництво «Політехніка», 2024, с. 190—193.

[2] Dmitry Khovratovich та Ivica Nikolić. «Rotational Cryptanalysis of ARX». В: *Fast Software Encryption*. За ред. Seokhie Hong та Tetsu Iwata. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, с. 333—346. ISBN: 978-3-642-13858-4.

[3] François-Xavier Standaert та ін. «SEA: A Scalable Encryption Algorithm for Small Embedded Applications». В: *Smart Card Research and Advanced Applications*. За ред. Josep Domingo-Ferrer, Joachim Posegga та Daniel Schreckling. Springer Berlin Heidelberg, 2006, с. 222—236. ISBN: 978-3-540-33312-8.

[4] Magnus Daum. «Cryptanalysis of Hash functions of the MD4-family». В: 2005. URL: https://api.semanticscholar.org/CorpusID:124174179.

[5] Dmitry Khovratovich та ін. «Rotational cryptanalysis of ARX revisited». В: *Fast Software Encryption: 22nd International Workshop, FSE 2015 Revised Selected Papers [Lecture Notes in Computer Science, Volume 9054]*. За ред. G Leander. Germany: Springer, 2015, с. 519—536. DOI: 10 . 1007 / 978 - 3 - 662 - 48116 - 5 \ _25. URL: https://eprints.qut.edu.au/101038/.

[6] Tomer Ashur та Yunwen Liu. «Rotational Cryptanalysis in the Presence of Constants». В: *IACR Transactions on Symmetric Cryptology* 2016.1

(груд. 2016), 57–70. DOI: 10.13154/tosc.v2016.i1.57-70. URL: https://tosc.iacr.org/index.php/ToSC/article/view/535.

[7] Pawee Morawiecki, Josef Pieprzyk та Marian Srebrny. «Rotational Cryptanalysis of Round-Reduced Keccak». В: лип. 2014. ISBN: 978-3-662-43932-6. DOI: 10.1007/978-3-662-43933-3_13.

[8] Liliya Kraleva, Tomer Ashur та Vincent Rijmen. «Rotational Cryptanalysis on MAC Algorithm Chaskey». В: *IACR Cryptology ePrint Archive*. 2020. URL: https://api.semanticscholar.org/CorpusID:218625690.

[9] Gilles van Assche. *A Rotational Distinguisher on Shabal's Keyed Permutation and Its Impact on the Security Proofs*. 2010. URL: http://gva.noekeon.org/papers/ShabalRotation.pdf.

[10] Emmanuel Bresson та ін. *Shabal, a Submission to NIST's Cryptographic Hash Algorithm Competition*. Submission to NIST. 2008.

[11] Єгор Сергійович Панасюк. *Диференціально-обертальний криптоаналіз деяких ускладнюючих функцій ARX-криптосистем : дипломна робота ... бакалавра : 113 Прикладна математика*. 2023. URL: https://ela.kpi.ua/handle/123456789/63228.

ДОДАТОК А ТЕКСТИ ПРОГРАМ

А.1 Програма 1

Програмний код на мові C++ для обчислення статистики проходження пар обертання через розглянуті у роботі функції.

```
#include <iostream>
#include <iomanip>
#include <cstdint>
#include <fstream>

#define F1
#define F2

// #define PRINT_TABLE

uint16_t N_first = 32;
uint16_t N_last = 32;

uint16_t ROT_first = 1;
uint16_t ROT_last = 1;

uint16_t N;
uint16_t ROT;
uint64_t modulo;

uint64_t cycl_rot_left(uint64_t x)
{
    return ((x << ROT) % modulo) ^ (x >> (N - ROT));
};

#ifdef F1
    uint64_t f_1(uint64_t x) { return (3*x) % modulo; }
#endif

#ifdef F2
    uint64_t f_2(uint64_t x) { return (5*x) % modulo; }
#endif


#ifdef PRINT_TABLE
    struct pack_res
    {
        uint64_t x;
```

```
    uint64_t f_x;
    uint64_t f_x_r;
    uint64_t xr;
    uint64_t f_xr;
};

void calculate_results(pack_res results[], size_t amnt, auto f = f_1)
{
    for(uint64_t i = 0; i < amnt; ++i)
    {
        results[i].x = i;
        results[i].f_x = f(i);
        results[i].f_x_r = cycl_rot_left(results[i].f_x);
        results[i].xr = cycl_rot_left(i);
        results[i].f_xr = f(results[i].xr);
    }
}

uint64_t print_table(pack_res results[], size_t amnt, std::ostream& write_st,
                     uint16_t width)
{
    uint64_t stat = 0;

    write_st << std::setw(width) << "x" << '\t'
            << std::setw(width) << "f(x)" << '\t'
            << std::setw(width) << "x <<< r" << '\t' << " | "
            << std::setw(width) << "f(x) <<< r" << '\t'
            << std::setw(width) << "f(x <<< r)" << '\n';
    write_st << std::setfill('0');

    for(uint64_t i = 0; i < amnt; ++i)
    {
        write_st << std::setw(width) << std::bitset<5>(results[i].x) << '\t'
                << std::setw(width) << std::bitset<5>(results[i].f_x) << '\t'
                << std::setw(width) << std::bitset<5>(results[i].xr) << '\t' << " | "
                << std::setw(width) << std::bitset<5>(results[i].f_x_r) << '\t'
                << std::setw(width) << std::bitset<5>(results[i].f_xr) << '\t';

        if(results[i].f_x_r == results[i].f_xr)
        {
            write_st << "+";
            stat++;
        }
        write_st << "\n";
    }
```

```
        write_st << std::setfill('␣');
        return stat;
    }
#else
    uint64_t calculate_stat_only(size_t amnt, auto f)
    {
        uint64_t stat = 0;

        for(uint64_t x = 0; x < amnt; ++x)
            if(cycl_rot_left(f(x)) == f(cycl_rot_left(x)))
                stat++;

        return stat;
    }
#endif

uint64_t calculate_theoretical_f1(uint64_t n, uint64_t r)
{
    uint64_t mult1 = (1U << (n-r)) + ((n-r) % 2 ? 1 : 2);
    uint64_t mult2 = (1U << r) + (r % 2 ? 1 : 2);

    if(mult1 % 9 == 1)
    {
        mult1 /= 9;
    }
    else if(mult2 % 9 == 1)
    {
        mult2 /= 9;
    }
    else
    {
        mult1 /= 3;
        mult2 /= 3;
    }

    uint64_t th_stat = mult1 * mult2;
    return th_stat;
}

uint64_t calculate_theoretical_f2(uint64_t n, uint64_t r)
{
    uint64_t th_stat = (uint64_t)3 * (uint64_t)(1U << (n-1)) +
                       (uint64_t)(((n-1) / 2) % 2 ? -1 : 1) * (3 + (n % 2 ? -1 : 1));
    return th_stat / (uint64_t)5;
```

```
}


int main(int argc, char *argv[])
{
    std::cout;
    if(argc == 5)
    {
        N_first = std::stoi(argv[1]);
        N_last = std::stoi(argv[argc - 2]);
        ROT_first = std::stoi(argv[argc - 1]);
        ROT_last = std::stoi(argv[argc - 1]);
    }


    for(uint16_t n = N_first; n <= N_last; ++n)
    {
        N = n;
        modulo = (uint64_t)1 << N;
#ifdef PRINT_TABLE
        pack_res* Results = new pack_res[modulo];
#endif

        for(uint16_t r = ROT_first; r <= ROT_last; ++r)
        {
            ROT = r;
// ---------------------------------------- F1 ----------------------------------------
#ifdef F1
            std::ofstream result_log_1("prints_f1.txt", std::ios_base::app);
            result_log_1 << "\n---------- 3x for (N, r) = (" << N
<< ", " << ROT << ") ----------\n";
#ifdef PRINT_TABLE
            calculate_results(Results, modulo, f_1);
            uint64_t stat = print_table(Results, modulo, result_log_1, N);
#else
            uint64_t stat = calculate_stat_only(modulo, f_1);
#endif
            result_log_1 << std::setw(20) << std::left << "Values:" << modulo << '\n'
                         << std::setw(20) << std::left << "Prob:"
<< std::setw(15) << double(stat) / double(modulo) << '\n'
                         << std::setw(20) << std::left << "Statistic:"    << stat   << '\n'
                         << std::setw(20) << std::left << "Theoretical:"
                         << calculate_theoretical_f1(N, ROT) << '\n';
#endif
```

```
// ---------------------------------------- F2 ----------------------------------------
#ifdef F2
            std::ofstream result_log_2("prints_f2.txt", std::ios_base::app);
            result_log_2 << "\n------------------------ 5x for (N, r) = (" << N
<< ", " << ROT << ") ------------------------\n";
#ifdef PRINT_TABLE
            calculate_results(Results, modulo, f_2);
            uint64_t stat = print_table(Results, modulo, result_log_2, N);
#else
            uint64_t stat = calculate_stat_only(modulo, f_2);
#endif
            result_log_2 << std::setw(20) << "Values:" << modulo << '\n'
                         << std::setw(20) << "Prob:"
<< std::setw(15) << double(stat) / double(modulo) << '\n'
                         << std::setw(20) << "stat:"    << stat   << '\n'
                         << std::setw(20) << "Theoretical:"
                         << calculate_theoretical_f2(N, ROT) << '\n';
#endif
        }


#ifdef PRINT_TABLE
        delete[] Results;
#endif
    }
}
```

ДОДАТОК Б ТАБЛИЦІ

Таблиця Б.1 – Порівняння, зроблене Ховратовичем та Ніколічем у роботі [5], для демонстрації різниці між оцінками, побудованими при використанні припущення марковості та при його відсутності.

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