УДК 004.056.55:512.6

ДИФЕРЕНЦІАЛЬНИЙ КРИПТОАНАЛІЗ ОДНІЄЇ З УСКЛАДНЮЮЧИХ ФУНКЦІЙ ГЕШ-ФУНКЦІЇ SHABAL

П. О. Бондар[1,a]

[1] Навчально-науковий Фізико-технічний інститут

Анотація

У даній роботі досліджуються диференціальні ймовірності функції модульного множення на 3. Ця функція використовується, зокрема, у псевдовипадковій перестановці геш-функції Shabal.

Одержано аналітичні умови існування вхідних значень, для яких виконується умова диференціальної ймовірності, та алгоритм обчислення ймовірності проходження диференціальної пари через функцію для довільних різниць α та β. Обчислення результату відбувається ефективно, за $O(n)$ операцій, та потребує не більше ніж одного паралельного обходу векторів α та β.

Ключові слова: ARX-криптосистеми, геш-функція Shabal, диференціальний криптоаналіз

Вступ

ARX-криптосистеми використовують дуже обмежений набір операцій: модульне додавання, двійкові зсуви та побітове додавання. Завдяки простоті цих операцій, такі криптопримітиви здобули значного поширення в системах, що мають обмежені обчислювальні ресурси та вимагають ефективних алгоритмів.

У класичному диференціальному криптоаналізі розглядаються різниці за операцією $\oplus$. Це є доцільним в контексті ARX-криптосистем, оскільки циклічний та нециклічний зсуви, а також саме побітове додавання є лінійними відносно цієї операції. Головна складність виникає при оцінці ймовірностей проходження $\oplus$-різниць через функцію модульного додавання.

Об'єктом дослідження виступає відображення $f(x) = 3x \mod 2^n$, що свого часу було використано для виконання псевдовипадкової перестановки геш-функції Shabal [1]. У випадку функції $f(x)$ модульне додавання присутнє, хоч і не явно.

Попередньо вже було встановлено обертальні властивості цієї функції [2], а в роботі [3] було побудовано оцінки диференціальних ймовірностей для LRX-аналогів функції $f(x)$, в яких замість модульного додавання використовувалися різні логічні операції. Наведений в цій роботі метод дозволить ефективно обчислювати ймовірності з точки зору диференціального криптоаналізу.

1. Терміни та позначення

У даній роботі використовуються такі позначення:

- $V_n = \{\mathtt{0}, \mathtt{1}\}^n$ — множина двійкових векторів довжини n;
- $x = (x_{n-1}, x_{n-2}, \dots, x_{\mathtt{0}}) \in V_n$ — двійковий вектор довжини n;
- $\oplus$ — операція двійкового додавання (XOR);
- $eq(x, y, z, \dots)$ — двійковий вектор, що має значення 1 на позиціях, на яких збігаються значення всіх вхідних векторів, та значення $\mathtt{0}$ на всіх інших позиціях;
- $carry(x, y)$ — вектор бітів переносу, що утворюються при модульному додаванні векторів $x, y \in V_n$,
- $\langle a, b, c\rangle$ — функція мажоризації, що повертає 1, якщо щонайменше два біти з $a, b, c \in \{\mathtt{0}, \mathtt{1}\}$ набувають значення 1, та $\mathtt{0}$ в протилежному випадку.
- $\overline{x}$ — побітове заперечення вектора $x \in V_n$.

Введемо також позначення диференціальної ймовірності функції $f(x)$ для довільних різниць $\alpha, \beta \in V_n$:

$$xdp^f(\alpha \to \beta) = \Pr_x\{f(x \oplus \alpha) = f(x) \oplus \beta\}.$$

У 2001 році Ліпмаа та Моріаі [4] запропонували ефективний алгоритм обчислення диференціальних ймовірностей для функції додавання двох змінних за модулем. Побудова оцінки ґрунтувалася на співставленні та аналізі бітів переносу, що утворюються внаслідок додавання пар векторів (x, y) та $(x \oplus \alpha, y \oplus \beta)$. Отриманий ними результат полягає у двох етапах побудови оцінки.

1. **Перевірка на неможливість**, що дозволяє визначити чи є загалом сумісними умови ймовірності за обраних значень різниць $\alpha, \beta, \gamma \in V_n$.
2. **Обчислення ймовірності**, що полягає в перевірці бітової структури векторів α, β, γ та побудові оцінки залежно від неї.

Цей результат показав, що для оцінки диференціальних ймовірностей не є обов'язковим повний перебір всіх можливих вхідних векторів, та дозволив

[a]bondar.petro@lll.kpi.ua

Таблиця 1. Двійкове представлення операції додавання для $f(x) = 3x \bmod 2^n$.

c	c_{n-1}	c_{n-2}	c_{n-3}	...	c_{i+1}	c_i	c_{i-1}	...	c_2	$c_1 = 0$	$c_0 = 0$
x	x_{n-1}	x_{n-2}	x_{n-3}	...	x_{i+1}	x_i	x_{i-1}	...	x_2	x_1	x_0
$2x$	x_{n-2}	x_{n-3}	x_{n-4}	...	x_i	x_{i-1}	x_{i-2}	...	x_1	x_0	0
$3x$	l_{n-1}	l_{n-2}	l_{n-3}	...	l_{i+1}	l_i	l_{i-1}	...	l_2	l_1	l_0

вирішити низку супровідних задач, пов'язаних із аналізом ARX-криптосистем.

2. Диференціальний криптоаналіз функції модульного множення на три

Функція $f(x) = 3x \bmod 2^n$ за своєю структурою схожа на звичайну функцію додавання за модулем, яку розглядали Ліпмаа та Моріаі, оскільки її можна представити як $f(x) = x + (x \ll 1) \bmod 2^n$. Проте, насправді залежність між бітами вхідного значення сильно впливає на значення диференціальних ймовірностей, що не дозволяє використовувати вже отриманий результат, навіть із незначними змінами.

Загальна ж ідея побудови оцінки для $f(x)$ також полягатиме в тому, щоб використати інформацію, що зберігається у векторі бітів переносу при додаванні x та $x \ll 1$. Розглянемо двійкове представлення цієї операції позначивши вектор $l = 3x \bmod 2^n$, наведене у таблиці 1.

У наведеній таблиці c_i позначають біти вектору $c = carry(x,\, x \ll 1)$, які окремо можна обчислити за допомогою рекуренти:

$$\begin{gathered} c_0 = c_1 = 0 \\ \forall i \in \{2, \dots, n-1\} : \; c_i = \langle x_{i-1}, x_{i-2}, c_{i-1} \rangle. \end{gathered} \tag{1}$$

Позначимо також $c' = carry(x \oplus \alpha,\, (x \oplus \alpha) \ll 1)$ – вектор бітів переносу операції додавання, що виникає при обчисленні значення $f(x \oplus \alpha)$. Оскільки побітове додавання вектора α тут ніяк не впливає на обчислення модульного додавання, то біти c'_i обчислюються за таким же принципом.

Власне значення бітів результату обчислюються таким чином:

$$\forall i \in \{0, \dots, n-1\} : \; l_i = x_i \oplus x_{i-1} \oplus c_i. \tag{2}$$

Сформулюємо декілька допоміжних тверджень, необхідних для подальшого обчислення диференціальних ймовірностей $xdp^{3x}(\alpha \to \beta)$.

В межах алгоритму обчислення такої диференціальної ймовірності виникає потреба в обчисленні ймовірності виконання умови рівності біту вхідного вектора та наступного біту переносу. Позначимо цю ймовірність $q_i = \Pr\{x_{i-1} = c_i\}$.

Твердження 1. Для довільного випадкового вектора $x = (x_{n-1}, \dots, x_0)$ та вектора бітів переносу $c = carry(x,\, x \ll 1)$, а також довільного $i \in \{2, \dots, n-1\}$ виконується:

$$q_i = 1 - \frac{1}{2} q_{i-1}.$$

Доведення. Розглянемо ймовірність

$$\Pr\{x_{i-1} \neq c_i\} = 1 - \Pr\{x_{i-1} = c_i\}.$$

Умова цієї ймовірності породжує два можливих варіанти.

1. Якщо $x_{i-1} = 0$, тоді з визначення вектору бітів переносу (1), за умови $x_{i-1} \neq c_i$, має виконуватись $x_{i-2} = c_{i-1} = 1$.
2. Якщо $x_{i-1} = 1$, тоді з аналогічних міркувань повинна виконуватись рівність $x_{i-2} = c_{i-1} = 0$.

Ці два випадки є незалежними між собою, а отже ймовірність повної події буде дорівнювати сумі цих двох ймовірностей:

$$\begin{aligned} &\Pr\{x_{i-1} \neq c_i\} = \\ &= \Pr\{x_{i-1} = 0\} \cdot \Pr\{x_{i-2} = c_{i-1} = 1\} + \\ &\qquad + \Pr\{x_{i-1} = 1\} \cdot \Pr\{x_{i-2} = c_{i-1} = 0\} = \\ &= \frac{1}{2} \Pr\{x_{i-2} = c_{i-1} = 1\} + \frac{1}{2} \Pr\{x_{i-2} = c_{i-1} = 0\} = \\ &= \frac{1}{2} (\Pr\{x_{i-2} = c_{i-1} = 1\} + \Pr\{x_{i-2} = c_{i-1} = 0\}) = \\ &\qquad = \frac{1}{2} \Pr\{x_{i-2} = c_{i-1}\}. \end{aligned}$$

Власне із цього легко отримати необхідне:

$$\Pr\{x_{i-1} = c_i\} = 1 - \frac{1}{2} \Pr\{x_{i-2} = c_{i-1}\} = 1 - \frac{1}{2} q_{i-1}.$$

Що і потрібно було довести. □

Представимо диференціальну ймовірність для довільних різниць α та β через значення $\Delta c = (\delta_{n-1}, \dots, \delta_0) = c \oplus c'$.

Лема 1. Для довільних $\alpha, \beta \in V_n$, диференціальна ймовірність для $f(x) = 3x \bmod 2^n$ обчислюється так:

$$xdp^{3x}(\alpha \to \beta) = \sum_{x \in V_n} [c' \oplus c = \alpha \oplus (\alpha \ll 1) \oplus \beta].$$

Доведення. Скористаємось прямим визначенням

Таблиця 2. Варіанти, за якими обчислюватиметься ймовірність рівності на кожному біті, в залежності від значень α та β, для довільного $i \in \{1, \dots, n-2\}$.

№	α_i	α_{i-1}	δ_i	β_i	δ_{i+1}	Умова для біту $i+1$ за лемою 1	Ймовірність біту $i+1$
1	0	0	0	0	0	$\emptyset = \alpha_{i+1} \oplus \beta_{i+1}$	$[\alpha_{i+1} = \beta_{i+1}]$
2	0	0	1	1	$x_i \oplus x_{i-1}$	$x_i \oplus x_{i-1} = \alpha_{i+1} \oplus \beta_{i+1}$	1/2
3	0	1	0	1	$x_i \oplus c_i$	$x_i \oplus c_i = \alpha_{i+1} \oplus \beta_{i+1}$	1/2
4	0	1	1	0	$x_{i-1} \oplus c_i \oplus 1$	$x_{i-1} \oplus c_i = \alpha_{i+1} \oplus \beta_{i+1} \oplus 1$	?
5	1	0	0	1	$x_{i-1} \oplus c_i$	$x_{i-1} \oplus c_i = \alpha_{i+1} \oplus \beta_{i+1} \oplus 1$	?
6	1	0	1	0	$x_i \oplus c_i \oplus 1$	$x_i \oplus c_i = \alpha_{i+1} \oplus \beta_{i+1}$	1/2
7	1	1	0	0	$x_i \oplus x_{i-1} \oplus 1$	$x_i \oplus x_{i-1} = \alpha_{i+1} \oplus \beta_{i+1}$	1/2
8	1	1	1	1	1	$\emptyset = \alpha_{i+1} \oplus \beta_{i+1}$	$[\alpha_{i+1} = \beta_{i+1}]$

бітів результату додавання (2):

$$\begin{aligned} &xdp^{3x}(\alpha \to \beta) = \\ &\quad = \Pr\{(x \oplus \alpha) + 2(x \oplus \alpha) = (x + 2x) \oplus \beta\} = \\ &= \overline{\sum_{x \in V_n}} \Big[c' \oplus (x \oplus \alpha) \oplus ((x \oplus \alpha) \ll 1) = \\ &\qquad\qquad = c \oplus x \oplus (x \ll 1) \oplus \beta\Big] = \\ &= \overline{\sum_{x \in V_n}} \Big[c' \oplus \underline{x} \oplus \alpha \oplus \underline{(x \ll 1)} \oplus (\alpha \ll 1) = \\ &\qquad\qquad = c \oplus \underline{x} \oplus \underline{(x \ll 1)} \oplus \beta\Big] = \\ &\qquad = \overline{\sum_{x \in V_n}} \Big[c' \oplus c = \alpha \oplus (\alpha \ll 1) \oplus \beta\Big]. \end{aligned}$$

Лему доведено. □

Таке представлення диференціальної ймовірності дозволяє спростити аналіз можливих вхідних значень функції та звести його тільки до накладання окремих умов на значення векторів бітів переносу c та c'.

Наслідок 1. Довільний вхід $x \in V_n$ задовольняє умову тоді й тільки тоді, коли вектора Δc, отриманого внаслідок операцій модульного додавання, виконується

$$\delta_\emptyset = \alpha_\emptyset \oplus \beta_\emptyset;$$
$$\forall i \in \{1, \dots, n-1\} : \ \delta_i = \alpha_i \oplus \alpha_{i-1} \oplus \beta_i.$$

Перейдемо до побудови алгоритму обчислення $xdp^{3x}(\alpha \to \beta)$.

Розглянемо окремо значення бітів переносу c та c'. Скористаємось їх визначенням (1) та алгебраїчною нормальною формою функції мажоризації. Для довільного $i \geq 2$ буде виконуватись:

$$c_{i+1} = \langle x_i, x_{i-1}, c_i \rangle = x_i x_{i-1} \oplus x_i c_i \oplus x_{i-1} c_i.$$

З іншого боку, для c' буде виконуватись:

$$\begin{aligned} &c'_{i+1} = \langle x_i \oplus \alpha_i, x_{i-1} \oplus \alpha_{i-1}, c'_i \rangle = \\ &= (x_i \oplus \alpha_i)(x_{i-1} \oplus \alpha_{i-1}) \oplus c'_i (x_i \oplus \alpha_i) \oplus c'_i (x_{i-1} \oplus \alpha_{i-1}) = \\ &\quad = x_i x_{i-1} \oplus x_i \alpha_{i-1} \oplus x_{i-1} \alpha_i \oplus \alpha_i \alpha_{i-1} \oplus \\ &\qquad \oplus x_i c'_i \oplus \alpha_i c'_i \oplus x_{i-1} c'_i \oplus \alpha_{i-1} c'_i \end{aligned}$$

Додавши ці два значення, отримаємо значення бітів вектора Δc:

$$\begin{aligned} &\delta_{i+1} = c_{i+1} \oplus c'_{i+1} = \\ &= x_i (c_i \oplus \alpha_{i-1} \oplus c'_i) \oplus x_{i-1} (c_i \oplus \alpha_i \oplus c'_i) \oplus \\ &\qquad \oplus \alpha_i \alpha_{i-1} \oplus \alpha_i c'_i \oplus \alpha_{i-1} c'_i = \\ &= x_i (\delta_i \oplus \alpha_{i-1}) \oplus x_{i-1} (\delta_i \oplus \alpha_i) \oplus \langle \alpha_i, \alpha_{i-1}, \delta_i \oplus c_i \rangle. \end{aligned} \tag{3}$$

А оскільки значення перших двох бітів переносу завжди буде дорівнювати $\emptyset$, то і $\delta_\emptyset = \delta_1 = \emptyset$.

Користуючись отриманою інформацією про значення бітів Δc, а також умовою з леми 1, можна сформулювати обмеження на значення перших двох бітів векторів α і β:

$$\alpha_\emptyset \oplus \beta_\emptyset = \emptyset; \tag{4}$$
$$\alpha_1 \oplus \alpha_\emptyset \oplus \beta_1 = \emptyset. \tag{5}$$

Співставлення ж усіх інших значень $\alpha, \beta, \Delta c$ та умов, які вони накладають на значення бітів вхідного вектора x, наведено у таблиці 2. Значення δ_{i+1} в цій таблиці обчислюється за допомогою рекуренти (3), а умова обмеження отримана із наслідку 1.

Оскільки наведені перетворення дозволили повністю абстрагуватися від значень бітів переносу c', тоді доцільно визначити, які обмеження накладаються на вхідний вектор $x \in V_n$ та на біти переносу c. Розглядатимемо позиції $i = \overline{2 \dots n}$ в порядку зростання індексу, а на кожному наступному індексі припускаємо, що умова з таблиці 2 на попередньому індексі виконується.

Одразу можна помітити, що для випадків 1 та 8 (таблиця 2), умова не залежить від бітів вектора x, а це означає, що в цих двох випадках ймовірність завжди дорівнює 1, за умови $\alpha_{i+1} = \beta_{i+1}$. Тому це обмеження впливає тільки на умову можливості. Для випадків 2, 3, 6, 7 достатньо зафіксувати, щоб біт x_i дорівнював значенню, що задовольняло б рівність. А оскільки на попередніх кроках ніяким чином значення x_i не було зафіксоване і воно є незалежним зі значеннями c_i та x_{i-1}, то ймовірність цієї події становитиме строго $\frac{1}{2}$.

Лема 2. Для довільних α, β та довільного індексу $i \in \{1, \dots, n-2\}$, якщо $\alpha_i = \alpha_{i-1} = 1 \oplus \beta_i$ (випадки 2 та 7 з таблиці 2) або $1 \oplus \alpha_i = \alpha_{i-1} = \beta_i$ (випадки 3 та 6), тоді ймовірність виконання відповідної умови з таблиці 2 для біта $i+1$ становить $\frac{1}{2}$.

Також зауважимо, що умови 1, 4, 5 і 8 не накладають жодних обмежень на значення x_i, проте варіанти 4 і 5 накладають вимогу на рівність, або нерівність, в залежності від значень α_{i+1} та β_{i+1}, бітів x_{i-1} та c_i. Основна складність полягає в оцінці ймовірностей саме для цих випадків.

Розглядатимемо можливі варіанти із таблиці 2, як стани певної системи, тоді розглянемо можливі переходи між станами при зміні індексу $i \to i+1$. З огляду на вигляд бітів вектора α, а також додаткові обмеження в умовах 1 та 8, маємо такі переходи:

$$\begin{aligned} &1 \to 1,5; && 5 \to 3,4,7,8;\\ &2 \to 1,2,5,6; && 6 \to 3,4,7,8;\\ &3 \to 1,2,5,6; && 7 \to 3,4,7,8;\\ &4 \to 1,2,5,6; && 8 \to 4,8. \end{aligned}$$

Зауваження. В стан 5 можна потрапити тільки зі станів 1, 2, 3, 4, а в стан 4 – зі станів 5, 6, 7, 8. З послідовності ж станів 1 чи 8, можна вийти тільки в стани 5 чи 4 відповідно.

Але виникає питання виконання умови, що відповідає станам 4 та 5, при переході з іншого стану.

Твердження 2. При переході в стан 4 (на позиції i) зі станів 5, 6 та 7 імовірність виконання умови для наступного біта дорівнює 1, якщо $\alpha_{i+1} \neq \beta_{i+1}$, та 0, у протилежному випадку.

Доведення. Розглянемо ці випадки окремо для стану 4 ($\alpha_i = \beta_i = \emptyset$).

- **Перехід 7 → 4.**
  Зі стану 7 ми отримали умову, що
  $$x_{i-1} \oplus x_{i-2} = \alpha_i \oplus \beta_i = \emptyset,$$
  з чого випливає $x_{i-1} = x_{i-2}$, а тому виконується рівність
  $$c_i = \langle \underline{x_{i-1}}, \underline{x_{i-2}}, c_{i-1} \rangle = x_{i-1}.$$
- **Перехід 6 → 4.**
  Зі стану 6 умова має такий вид:
  $$x_{i-1} \oplus c_{i-1} = \alpha_i \oplus \beta_i = \emptyset,$$
  з чого слідує рівність $x_{i-1} = c_{i-1}$, а далі аналогічно отримуємо:
  $$c_i = \langle \underline{x_{i-1}}, x_{i-2}, \underline{c_{i-1}} \rangle = x_{i-1}.$$
- **Перехід 5 → 4.**
  У цьому випадку зі стану 5 ми отримали умову
  $$x_{i-2} \oplus c_{i-1} = \alpha_i \oplus \beta_i \oplus 1 = 1,$$
  а це означає, що $x_{i-2} = 1 \oplus c_{i-1}$. З цього випливає:
  $$c_i = \langle x_{i-1}, c_{i-1} \oplus 1, c_{i-1} \rangle = x_{i-1}.$$

В усіх трьох випадках ми отримали, що $c_i = x_{i-1}$, але за умовою стану 4 це виконується тоді й тільки тоді, коли виконується рівність $\alpha_{i+1} \oplus \beta_{i+1} \oplus 1 = \emptyset$, або ж $\alpha_{i+1} \neq \beta_{i+1}$. □

Аналогічно можна показати і для стану 5. Перехід $7 \to 4$ відповідатиме переходу $2 \to 5$, $6 \to 4$ – переходу $3 \to 5$, а $5 \to 4$ – відповідно переходу $4 \to 5$.

Твердження 3. При переході в стан 5 (на позиції i) зі станів 2, 3 та 4 ймовірність виконання умови для наступного біта дорівнює 1, якщо $\alpha_{i+1} \neq \beta_{i+1}$, та 0, у протилежному випадку.

Користуючись схожими міркуваннями, бо як у випадку зі станами 4 та 5, стани 1 та 8 так само мають рівні значення $\alpha_i = \beta_i$, можна сформулювати таке твердження.

Твердження 4. При потраплянні у стан 1 (або 8) із будь-якого стану, окрім 1 (або 8), автоматично виконується умова $x_{i-1} = c_i$ із ймовірністю 1.

Це твердження дозволяє визначити кінцеве значення рекуренти, що наведена у твердженні 1.

Оскільки самі по собі умови 1 та 8 не накладають жодних конкретних вимог на значення бітів вектора x, тоді у випадку, якщо така послідовність закінчується умовами 5 та 4 відповідно, проміжні значення x_i можна обирати таким чином, щоб врешті виконалася рівність $x_{i-1} \oplus c_i = \alpha_{i+1} \oplus \beta_{i+1} \oplus 1$.

Твердження 5. Якщо послідовність станів 1 закінчилася переходом у стан 5, тоді ймовірність виконання умови $x_{i-1} = c_i$ дорівнює значенню q_l (див. 1), де граничне значення $q_\emptyset = 1$, а l дорівнює кількості послідовних станів 1, що передують 5.

Аналогічно для переходу зі стану 8 у стан 4.

Тоді ймовірність виконання умови для станів 4 та 5 обчислюється так:

$$\Pr\{x_{i-1} \oplus c_i = \alpha_{i+1} \oplus \beta_{i+1} \oplus 1\} = \begin{cases} q_l, & \alpha_{i+1} \neq \beta_{i+1}, \\ 1 - q_l, & \alpha_{i+1} = \beta_{i+1}. \end{cases}$$

Зауваження. Для деяких значень α та β послідовність станів 1 може починатися нульового біта вектора x. Оскільки в цьому випадку значення $\Pr\{c_1 = x_\emptyset\} = \frac{1}{2}$, що алгебраїчно задовольняє визначення рекуренти: $q_1 = 1 - \frac{1}{2} q_\emptyset = \frac{1}{2}$. А отже, такий спосіб обчислення залишається застосовним.

Врешті, для рекуренти q_l можна отримати аналітичний вираз, щоб спростити обчислення.

Твердження 6. Для рекуренти із твердження 1 при граничному значенні $q_\emptyset = 1$ справедлива рівність

$$q_i = \frac{1}{3}\left(2 + (-2)^{-i}\right).$$

Отриманий набір спостережень повністю покриває всі можливі випадки, що виникають за різних значень вхідного вектора x та різниць α і β, завдяки чому можна сформулювати умову можливості та власне алгоритм обчислення ймовірності.

Для формулювання умов, за яких імовірність буде відмінною від нуля, необхідно ввести додаткові позначення векторів:

1) $v_{1,8} = ((eq(\alpha, \alpha \ll 1, \beta) \ll 1) \gg 2)$ – всі біти, що відповідають умовам 1 та 8, мають значення 1, при чому виключено перший біт та останній;
2) $v_{4,5} = ((eq(\alpha, \overline{\alpha \ll 1}, \beta) \ll 1) \gg 2)$ – аналогічний вектор для умов 4 та 5;
3) $\Delta = (\alpha \oplus \beta) \gg 2$ – вектор різниці векторів α та β, зсунутий на 2 праворуч для співставлення $i+1$ біту вектора до i вектора умови.

Теорема 1. *Для довільних значень $\alpha, \beta \in V_n$, значення $xdp^{3x}(\alpha \to \beta)$ набуває ненульового значення тоді й тільки тоді, коли одночасно виконуються такі чотири умови:*

1) $\alpha_0 = \beta_0$ (рівність 4);
2) $\alpha_1 \oplus \alpha_0 \oplus \beta_1 = 0$ (рівність 5);
3) $v_{1,8} \,\&\, \Delta = 0$ (умова випадків 1 та 8 з табл. 2);
4) $v_{4,5} \,\&\, (\overline{v_{1,8}} \ll 1) \,\&\, \overline{\Delta} = 0$ (за тв. 2 та 3).

Фактично, перевірка на можливість потребує лише виконання кількох двійкових операцій, що дозволяє ефективно перевірити, чи можливе взагалі проходження такого (α, β)-диференціала через функцію.

Алгоритм обчислення ймовірності

Вхід: алгоритм приймає значення $\alpha, \beta \in V_n$.

Вихід: алгоритм повертає ймовірність проходження диференціала $xdp^{3x}(\alpha \to \beta)$.

1. Перевірити умови можливості за теоремою 1. Якщо хоча б одна умова не виконується – повернути значення 0.
2. $P \leftarrow 1$.
3. Обчислити ймовірність для бітів, що відповідають умовам 2, 3, 6, 7 за допомогою леми 2:
$$P \leftarrow P \cdot 2^{-wt(v_{2,3,6,7})},$$
Тут $v_{2,3,6,7}$ – вектор бітів, що має значення 1 на позиціях, що відповідають варіантам 2, 3, 6, 7 з таблиці 2 (значення нульового та останнього бітів встановлені як 0):
$$v_{2,3,6,7} = (((eq(\alpha,\ \alpha \ll 1,\ \overline{\beta}) \oplus \\ \oplus\ eq(\overline{\alpha},\ \alpha \ll 1,\ \beta)) \ll 1) \gg 2).$$
4. Знайти всі послідовності станів 1, що переходять у стан 5, та всі послідовності станів 8, що переходять у стан 4.
Скласти масив пар $X \leftarrow [(i_0, l_0), \dots]$, де i_j – індекс стану 5 (або 4), l_j – довжина послідовності станів 1 (або 8).
5. Для кожного елемента $(i, l) \in X$ обчислити ймовірність виконання умови за допомогою твердження 5, а також аналітичного представлення рекуренти з твердження 6:
$$P \leftarrow P \cdot \begin{cases} \frac{1}{3}\left(2 + (-2)^{-l}\right), & \alpha_{i+1} \neq \beta_{i+1}, \\ 1 - \frac{1}{3}\left(2 + (-2)^{-l}\right), & \alpha_{i+1} = \beta_{i+1}. \end{cases}$$
6. Повернути значення ймовірності P.

Оскільки в ході побудови оцінки було розглянуто всі можливі варіанти переходів, то отримана за цим алгоритмом ймовірність буде точною оцінкою $xdp^{3x}(\alpha \to \beta)$. А завдяки фактично лінійній складності цього алгоритму обчислити цю ймовірність можна дуже ефективно.

Отримані результати було підтверджено шляхом комп'ютерного моделювання для різних довжин двійкових рядків n від 4 до 20. В усіх випадках результат обчислення за цим алгоритмом збігався з практичним результатом, отриманим шляхом прямого перебору.

Висновки

В межах даного дослідження було отримано ефективний алгоритм, що дозволяє обчислити диференціальні ймовірності для функції $f(x) = 3x \bmod 2^n$, при довільному значенні довжини векторів n та довільних різницях α та β. Наведений алгоритм має складність $O(n)$ з дуже малими константами, що дозволяє ефективно отримати оцінки ймовірностей для великих довжин векторів.

Надалі отриманий результат можна поєднати з оцінкою обертальних ймовірностей цієї функції та встановити оцінку для неї за диференціально-обертальним криптоаналізом. Також отриманий метод аналізу можна буде застосувати для побудови аналогічних оцінок функцій виду $(2^s + 1)x \bmod 2^n$.

Перелік використаних джерел

1. Shabal, a Submission to NIST's Cryptographic Hash Algorithm Competition / E. Bresson [et al.]. — 2008. — Submission to NIST.
2. *Бондар П. О.* Оцінювання стійкості ускладнюючих функцій спеціального виду до обертального криптоаналізу : дипломна робота ... бакалавра : 113 Прикладна математика / Бондар Петро Олександович. — 2024. — URL: https://ela.kpi.ua/handle/123456789/69271.
3. *Yakovliev S.* Differential properties of LRX-analogues of small constant multiplication // International Journal of Electronics and Telecommunications. — 2025. — Vol. vol. 71, No 1. — P. 95–100. — DOI: 10.24425/ijet.2025.153550. — URL: http://journals.pan.pl/Content/134699/PDF/13.pdf.
4. *Lipmaa H., Moriai S.* Efficient Algorithms for Computing Differential Properties of Addition. — 2001. — URL: https://eprint.iacr.org/2001/001. Cryptology ePrint Archive, Paper 2001/001.