2023 г.

Поиск невозможных дифференциалов для блочного шифра КБ256-3

Выполнили: Астраханцев Р.Г., Дмух. А.А., Астраханцева И.А.

Актуальность исследования

- Скорость зашифрования алгоритмом КБ256-3 более чем в 2.5 раза быстрее прямого зашифрования алгоритмом «Магма».
- Для алгоритма КБ256-3 сохраняются эксплуатационные характеристики, сопоставимые с алгоритмом «Магма».
- Эффективная реализация алгоритма КБ256-3 на платформах с ограниченными ресурсами (т.н. низкоресурсная криптография).

Цель исследования и основные задачи

Цель: поиск невозможных дифференциалов для шифра КБ256-3.
Задачи:

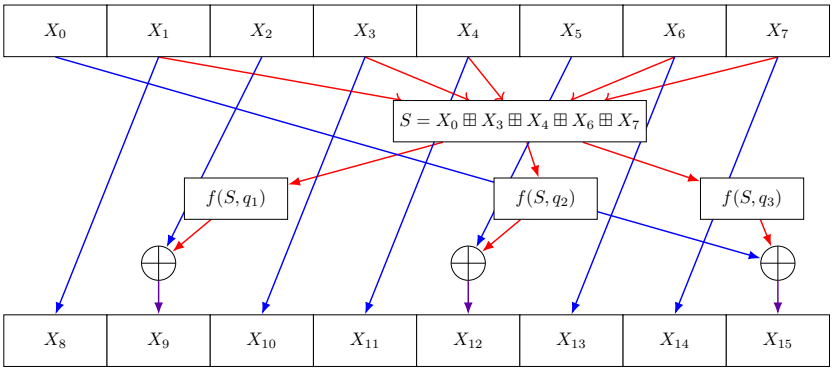
1. Разработка алгоритма поиска невозможных дифференциалов заданного вида для блочного шифра КБ256-3 (с заменой операции модульного сложения операцией XOR).

2. Поиск максимального числа итераций, на которое существуют невозможные дифференциалы, а также вид этих дифференциалов с использованием предложенного алгоритма.

3. Анализ результатов: оценка возможности поиска невозможных дифференциалов для шифра КБ256-3 с операцией модульного сложения.

Описание шифра КБ256-3

Преобразование КБ256-3 представляет из себя 16 раундов преобразования над текстом длины 256 бит.

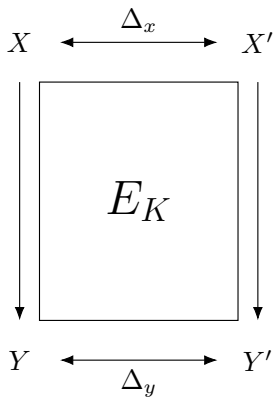


$f(S, q_i) = f_q(S \boxplus q_i)$,

f_q – последовательное применение подстановок $s_0, s_1, \ldots, s_7$ из ГОСТ Р 34.12–2015 «Магма» и циклического сдвига $T_{<<<19}$ на 19 позиций в сторону старших бит.

Первоначально в работе рассмотрена упрощённая версия шифра с заменой операции модульного сложения на XOR.

Метод невозможных дифференциалов



Дифференциал (Δ_x, Δ_y) называется невозможным, если $P(\Delta_x \to \Delta_y) = 0$.
Дифференциал (Δ_x, Δ_y) называется практически невозможным, если $P(\Delta_x \to \Delta_y) > 0$ и P мало.

С помощью невозможных дифференциалов можно проводить атаки по восстановлению ключа или различению блочного шифра от случайной перестановки.

В работе [1] был найден невозможный дифференциал для всех 16 итераций зашифрования вида

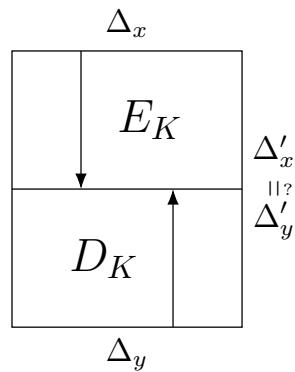
$$\Delta_x = (0, 2^{31}, 0, 0, 0, 0, 2^{31}, 0) \not\to (2^{31}, 0, 0, 2^{31}, 0, 0, 0, 0) = \Delta_y$$

[1] Fomichev V. M., Kurochkin A., Chuhno A. The difference relations and impossible differentials construction for the KB-256 algorithm // Прикладная дискретная математика. Приложение. — 2022. — No 15. — С. 73— 77.

Алгоритм нахождения невозможных дифференциалов

Без конкретизации параметров

1. Сформировать таблицу дифференциалов вероятности 1 для алгоритма зашифрования.
2. Сформировать таблицу дифференциалов вероятности 1 для алгоритма расшифрования.
3. **Для каждого** *входного и выходого дифференциала из соответствующей таблицы*
4. Сделать вывод относительно невозможности дифференциала в соответствии с критерием



Метод сокращений[1]

Для упрощённой версии шифра

$$L_1 = \begin{pmatrix} 0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \end{pmatrix}$$

$$L_2 = \begin{pmatrix} 0&0&0&0&0&0&0&0 \\ 0&1&0&0&1&0&0&1 \\ 0&0&0&0&0&0&0&0 \\ 0&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&0&1 \\ 0&0&0&0&0&0&0&0 \\ 0&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&0&1 \end{pmatrix}$$

$$L : V_{32}^8 \to V_{32}^8$$

$$L(x) = x \cdot L_1 \oplus F(x \cdot L_2)$$

$$\Downarrow$$

$$\widetilde{L} : V_2^8 \to V_2^8$$

$$\widetilde{L}(x) = x * (L_1 \oplus L_2)$$

$$|x| \leqslant |\widetilde{L}(x)|$$

$$\widetilde{L} = \begin{pmatrix} 0&0&0&0&0&0&0&1 \\ 1&1&0&0&1&0&0&1 \\ 0&1&0&0&0&0&0&0 \\ 0&1&1&0&1&0&0&1 \\ 0&1&0&1&1&0&0&1 \\ 0&0&0&0&1&0&0&0 \\ 0&1&0&0&1&1&0&1 \\ 0&1&0&0&1&0&1&1 \end{pmatrix}$$

$$\widetilde{L^{-1}} = \begin{pmatrix} 1&1&1&0&0&1&0&0 \\ 0&0&1&0&0&0&0&0 \\ 1&0&1&1&0&1&0&0 \\ 1&0&1&0&1&1&0&0 \\ 0&0&0&0&0&1&0&0 \\ 1&0&1&0&0&1&1&0 \\ 1&0&1&0&0&1&0&1 \\ 1&0&0&0&0&0&0&0 \end{pmatrix}$$

[1] Biham E., Biryukov A., Shamir A. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. — 1999.

Метод сокращений

Результаты для упрощённой версии шифра

Было найдено 25 классов невозможных дифференциалов на 6 раундов

$\widetilde{\Delta x}$	n_x	$\widetilde{\Delta y}$	n_y	n
$(\bullet, 0, 0, 0, 0, 0, 0, 0)$	3	$(0, 0, 0, 0, 0, 0, 0, \bullet)$	3	6
$(\bullet, 0, 0, 0, 0, 0, 0, 0)$	3	$(0, \bullet, 0, 0, 0, 0, 0, \bullet)$	3	6
$(\bullet, 0, 0, 0, 0, 0, 0, 0)$	3	$(0, 0, 0, 0, \bullet, 0, 0, 0)$	3	6
$(\bullet, 0, 0, 0, 0, 0, 0, 0)$	3	$(0, \bullet, 0, 0, 0, 0, 0, 0)$	3	6
$\vdots$				
$(0, 0, \bullet, 0, 0, 0, 0, 0)$	3	$(0, \bullet, 0, 0, 0, 0, 0, 0)$	3	6
$(0, 0, \bullet, 0, 0, 0, 0, 0)$	3	$(0, 0, 0, 0, \bullet, 0, 0, \bullet)$	3	6

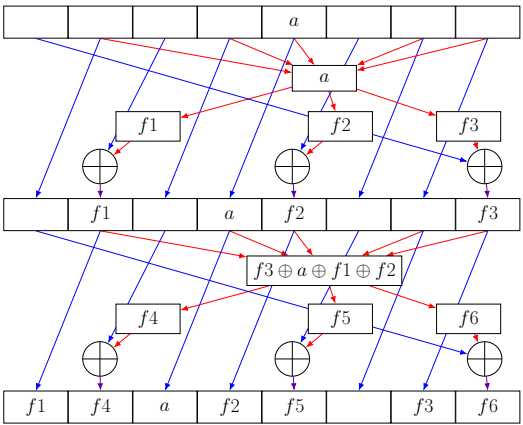
Другой способ группировать дифференциалы

Для упрощённой версии шифра

- Метод сокращений оперирует широкими классами дифференциалов (например, $\Delta_1 = (a, b, c, d, 0, 0, 0, 0)$ и $\Delta_2 = (a, a, b, b, 0, 0, 0, 0)$ неотличимы);
- Группируем дифференциалы покоординатно, помечая их ненулевые разности какой-то фиктивной переменной;
- Убираем дубликаты по замене переменных (например, $\Delta_1 = (a, a, b, b, 0, 0, 0, 0)$ и $\Delta_2 = (b, b, a, a, 0, 0, 0, 0)$);
- Искажения разности после прохождения через преобразования $f(s, q_1), f(s, q_2), f(s, q_3), \ldots$ помечаем новыми переменными $f_1, f_2, f_3, \ldots$ (для расшифрования $g_1, g_2, g_3, \ldots$);
- Распространяем дифференциалы до появления ненулевых элементов;
- Распространяем дифференциалы до появления элементов $f_1, f_2, f_3, g_1, g_2, g_3$.

Другой способ группировать дифференциалы

Демонстрация критерия с элементами $f_1, f_2, f_3, g_1, g_2, g_3$



В последнем раунде примера $S = f_1 \oplus f_2 \oplus f_3 \oplus a$. При этом a, f_1, f_2, f_3 являются не равными между собой ненулевыми величинами. Однозначно определить, будет ли S ненулевым, нельзя.

В примере видно, что $(0,0,0,0,a,0,0,0) \not\to (a,0,0,0,0,0,0,0)$, однако однозначно утверждать, что $(0,0,0,0,a,0,0,0) \not\to (0,a,0,0,0,0,0,0)$ нельзя.

Другой способ группировать дифференциалы

Результаты для упрощённой версии шифра

Невозможные дифференциалы

Δx	n_x	Δy	n_y	n
$(a, a, 0, 0, 0, 0, 0, a)$	7	$(a, a, a, 0, 0, 0, 0, 0)$	9	16
$(a, a, 0, 0, 0, 0, 0, a)$	9	$(a, a, a, 0, 0, 0, 0, 0)$	7	16
		$\vdots$		
$(a, 0, 0, 0, 0, 0, a, a)$	6	$(0, a, a, a, 0, 0, 0, 0)$	8	14

Практически невозможные дифференциалы

Δx	n_x	Δy	n_y	n
$(a, a, 0, 0, 0, 0, 0, a)$	9	$(a, a, a, 0, 0, 0, 0, 0)$	9	18
$(a, a, 0, 0, 0, 0, 0, a)$	9	$(a, 0, 0, a, 0, 0, 0, 0)$	8	17
		$\vdots$		
$(b, a, 0, b, 0, 0, a, b)$	5	$(a, a, a, 0, 0, 0, 0, 0)$	9	14

На 8-18 раундов обнаружено 14783 классов невозможных дифференциалов и 57041 классов практически невозможных дифференциалов.
На 14 раундов и более, соответственно, 43 и 395 классов.

Получение невозможных дифференциалов для модульного сложения

- Для элемента $e_{31} = 2^{31}$ результаты операций XOR и сложения по модулю 2^{32} с любым элементом из V_{32} совпадают;
- Из полученных диффереициалов нужно выбрать только те, в которых используется одинаковая входная разность, и заменить её на e_{31} (т.е. вида $(\Delta x_0, \Delta x_1, \Delta x_2, \Delta x_3, \Delta x_4, \Delta x_5, \Delta x_6, \Delta x_7)$, $\Delta x_i \in \{0, \alpha\}$ $\forall i = \overline{0,7}$ и принять $\alpha = e_{31}$);

Получение невозможных дифференциалов для модульного сложения

Результаты

- Таким образом получено 1778 невозможных дифференциалов и 3122 практически невозможных дифференциалов на 8 раундов и более.
- Среди найденных дифференциалов обнаружен дифференциал на 16 раундов из работы [1].
- Дифференциал на 18 раундов имеет максимальную длину, является невозможным (доказано отдельно) и записывается в виде

$$\Delta_x = (2^{31}, 2^{31}, 0, 0, 0, 0, 0, 2^{31}) \not\to (2^{31}, 2^{31}, 2^{31}, 0, 0, 0, 0, 0) = \Delta_y$$

[1] Fomichev V. M., Kurochkin A., Chuhno A. The difference relations and impossible differentials construction for the KB-256 algorithm // Прикладная дискретная математика. Приложение. — 2022. — No 15. — С. 73— 77.

Невозможные дифференциалы для КБ256-3

Δx	n_x	Δy	n_y	n	Тип
$(2^{31}, 2^{31}, 0, 0, 0, 0, 0, 2^{31})$	9	$(2^{31}, 2^{31}, 2^{31}, 0, 0, 0, 0, 0)$	9	18	I
$(2^{31}, 2^{31}, 0, 0, 0, 0, 0, 2^{31})$	9	$(2^{31}, 0, 0, 2^{31}, 0, 0, 0, 0)$	8	17	A
$(2^{31}, 0, 0, 0, 0, 0, 2^{31}, 2^{31})$	8	$(2^{31}, 2^{31}, 2^{31}, 0, 0, 0, 0, 0)$	9	17	A
$(2^{31}, 2^{31}, 0, 0, 0, 0, 0, 2^{31})$	9	$(0, 2^{31}, 2^{31}, 2^{31}, 0, 0, 0, 0)$	8	17	A
$(0, 2^{31}, 0, 0, 0, 0, 2^{31}, 0)$	8	$(2^{31}, 2^{31}, 2^{31}, 0, 0, 0, 0, 0)$	9	17	A
$(2^{31}, 2^{31}, 0, 0, 0, 0, 0, 2^{31})$	9	$(2^{31}, 2^{31}, 0, 2^{31}, 2^{31}, 0, 0, 0)$	7	16	A
$(2^{31}, 2^{31}, 0, 0, 0, 0, 0, 2^{31})$	7	$(2^{31}, 2^{31}, 2^{31}, 0, 0, 0, 0, 0)$	9	16	I
⋮					
$(0, 0, 2^{31}, 2^{31}, 2^{31}, 0, 0, 0)$	4	$(2^{31}, 2^{31}, 2^{31}, 0, 2^{31}, 0, 0, 2^{31})$	4	8	A
$(2^{31}, 0, 2^{31}, 2^{31}, 2^{31}, 2^{31}, 0, 0)$	4	$(0, 2^{31}, 0, 0, 2^{31}, 2^{31}, 2^{31}, 2^{31})$	4	8	A

Результаты

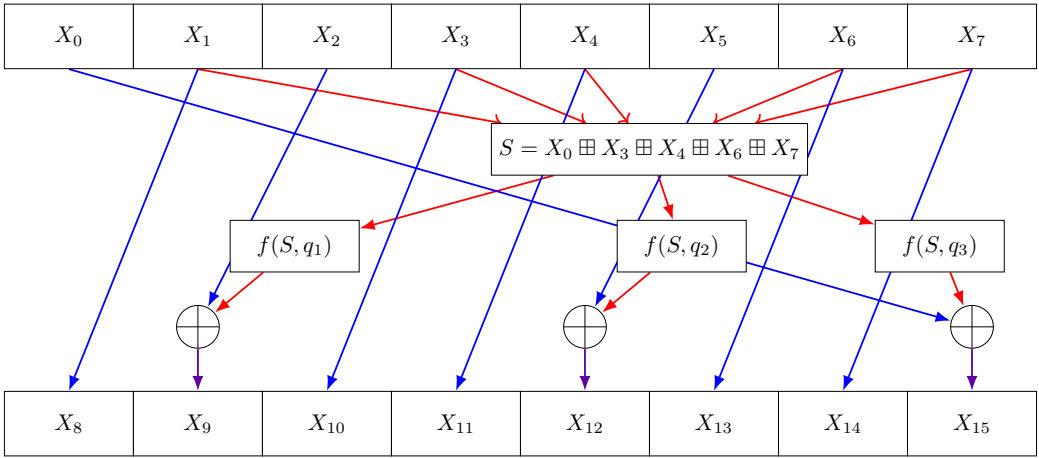
1. Построены невозможные и практически невозможные дифференциалы на 8-18 раундов для шифра КБ256-3 в количестве 1778 и 3122 штук соответственно.
2. Доказано наличие невозможного дифференциала на 18 итераций раундового преобразования шифра КБ256-3.
3. Было показано, что не всегда использование модульного сложения позволяет защититься от метода невозможных дифференциалов.

Рекомендации

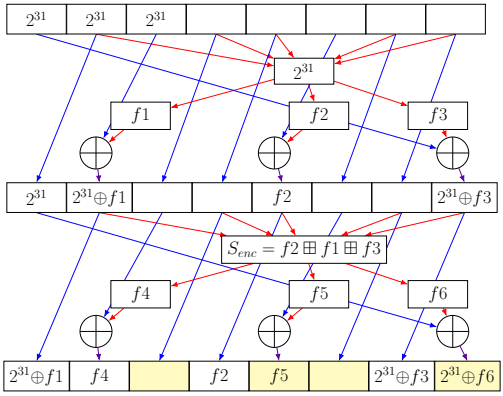
1. Для защиты от атак с использованием методов невозможных дифференциалов предлагается увеличить число применений раундового преобразования минимум на 6-10 итераций.

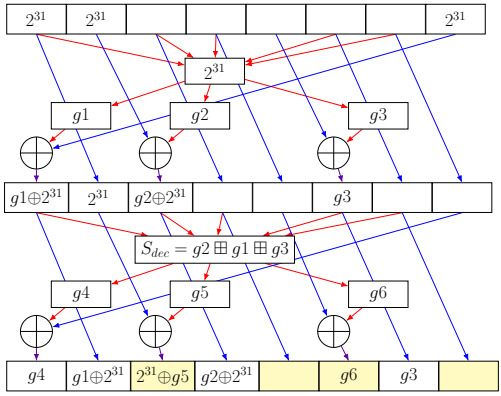
2. В качестве дополнительной защиты рекомендуется после вычисления S дополнительно использовать 1 раунд рассеивающего преобразования алгоритма «Магма» с циклическим сдвигом на 19 позиций и подавать результат на вход соответствующим F-блокам.

Раундовое преобразование КБ256-3



Часть доказательства невозможности дифференциала на 18 раундов





$$\Delta_x = (2^{31}, 2^{31}, 0, 0, 0, 0, 0, 2^{31}) \not\to (2^{31}, 2^{31}, 2^{31}, 0, 0, 0, 0, 0) = \Delta_y$$

Метод сокращений

Детали операции $*$

$$L_1 = \begin{pmatrix} 0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \end{pmatrix}$$

$$L_2 = \begin{pmatrix} 0&0&0&0&0&0&0&0 \\ 0&1&0&0&1&0&0&1 \\ 0&0&0&0&0&0&0&0 \\ 0&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&0&1 \\ 0&0&0&0&0&0&0&0 \\ 0&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&0&1 \end{pmatrix}$$

$$\widetilde{L} : V_2^8 \to V_2^8$$

$$\widetilde{L}(x) = x * (L_1 \oplus L_2)$$

$$|x| \leqslant |\widetilde{L}(x)|$$

$$* : V_2^8 \times M_{8\times 8} \to V_2^8$$

$$(\widehat{\Delta y_0}, \ldots, \widehat{\Delta y_7}) = \widehat{\Delta y} = \widehat{\Delta x} * \widehat{L}$$

$$\widehat{\Delta y_i} = \bigvee_{j=0}^{7} \widehat{\Delta x_j} \cdot \widehat{L_{ji}} \qquad \forall i = \overline{0,7}$$

$$\widetilde{L} = \begin{pmatrix} 0&0&0&0&0&0&0&1 \\ 1&1&0&0&1&0&0&1 \\ 0&1&0&0&0&0&0&0 \\ 0&1&1&0&1&0&0&1 \\ 0&1&0&1&1&0&0&1 \\ 0&0&0&0&1&0&0&0 \\ 0&1&0&0&1&1&0&1 \\ 0&1&0&0&1&0&1&1 \end{pmatrix}$$

$$\widetilde{L^{-1}} = \begin{pmatrix} 1&1&1&0&0&1&0&0 \\ 0&0&1&0&0&0&0&0 \\ 1&0&1&1&0&1&0&0 \\ 1&0&1&0&1&1&0&0 \\ 0&0&0&0&0&1&0&0 \\ 1&0&1&0&0&1&1&0 \\ 1&0&1&0&0&1&0&1 \\ 1&0&0&0&0&0&0&0 \end{pmatrix}$$