49 $L3$

Задача. В функции хэширования `Bash` используется преобразование $L3$. Оно ставит в соответствие тройке 64-разрядных слов (w_0, w_1, w_2) новую тройку (W_0, W_1, W_2), в которой

$$\begin{aligned} W_0 &= w_0 \oplus w_1 \oplus w_2, \\ W_1 &= w_1 \oplus w_0^8 \oplus W_0^{53}, \\ W_2 &= w_2 \oplus w_2^{14} \oplus (w_1 \oplus W_0^{53})^1. \end{aligned}$$

Здесь w^d — результат циклического сдвига слова w на d позиций в сторону старших разрядов. Постройте алгоритм обращения $L3$, т. е. определения (w_0, w_1, w_2) по (W_0, W_1, W_2). Постарайтесь использовать в алгоритме как можно меньше сложений $\oplus$ и сдвигов.

Решение. Обозначим $[m_1, n_1, m_2, n_2] = [8, 53, 14, 1]$, вместо $\oplus$ будем писать $+$, а вычисления со сдвигами вести по модулю 64.

Обратить $L3$ можно следующим образом:

1. Решить уравнение

$$w_2 + w_2^{m_1} + w_2^{m_2} + w_2^{m_1+m_2} + w_2^{m_1+n_2} = W_0^{m_1+n_2} + W_0^{m_1+n_1+n_2} + W_1^{n_2} + W_2 + W_2^{m_1}$$

   относительно w_2.

2. Определить $w_0 = W_1^{-m_1} + W_2^{-m_1-n_2} + w_2^{-m_1-n_2} + w_2^{m_2-m_1-n_2}$.

3. Определить $w_1 = W_0^{n_1} + W_2^{-n_2} + w_2^{-n_2} + w_2^{m_2-n_2}$.

Основная сложность — решение уравнения на шаге 1. Это уравнение можно записать следующим образом:

$$f(w_2) = F(W_0, W_1, W_2).$$

Здесь $f(x) = 1 + x^{n_1} + x^{m_2} + x^{m_1+m_2} + x^{m_1+n_2}$ — характеристический многочлен. Он описывает сумму сдвигов слова w_2 при его подстановке вместо формальной переменной x (моном $1 = x^0$ дает слово $w_2^0 = w_2$). Тождество $w_2^{i+64} = w_2^i$ означает, что мономы x^{i+64} и x^i эквиваленты, т. е. $f(x)$ можно считать элементом факторкольца $R = \mathbb{F}_2[x]/(x^{64} + 1)$.

Для решения уравнения следует найти в R мультипликативно обратный к f многочлен f^{-1} и применить его к правой части:

$$w_2 = f^{-1}(F(W_0, W_1, W_2)).$$

Многочлен f обратим в R тогда и только тогда, когда он содержит нечетное число мономов, т. е. $f(1) = 1$. При этом $f^{-1}(x) = (f(x))^{63}$.

Действительно, пусть $f(x) = \sum_{i=0}^{63} a_i x^i$ обратим, т. е. $\sum_i a_i = 1$. Тогда

$$f(x)^{64} = \sum_{i=0}^{63} a_i \left(x^{64}\right)^i \equiv \sum_{i=0}^{63} a_i = 1 \pmod{x^{64} + 1}.$$

Это значит, что в кольце R произведение $f \cdot f^{63}$ дает 1, т. е. $f^{-1} = f^{63}$.

Подставляя вместо $[m_1, n_1, m_2, n_2]$ заданные значения, получаем

$$f(x) = 1 + x^8 + x^9 + x^{14} + x^{22}.$$

Обратный многочлен

$$\begin{aligned} f^{-1}(x) = {} & x^4 + x^5 + x^6 + x^8 + x^9 + x^{11} + x^{13} + x^{14} + x^{18} + \\ & + x^{19} + x^{21} + x^{22} + x^{25} + x^{28} + x^{29} + x^{30} + \\ & + x^{32} + x^{33} + x^{34} + x^{36} + x^{41} + x^{42} + x^{43} + \\ & + x^{44} + x^{45} + x^{46} + x^{48} + x^{49} + x^{51} + x^{52} + \\ & + x^{54} + x^{55} + x^{56} + x^{57} + x^{58} + x^{60} + x^{61}. \end{aligned}$$

Он содержит 37 мономов.

Подсчитаем сложность обращения $L3$:

1. Правую часть уравнения на шаге 1 можно вычислить за 4 сдвига и 4 сложения.
2. Найти w_2, применяя f^{-1}, можно за 37 сдвигов и 36 сложений.
3. Правую часть выражения для w_0 можно вычислить за 3 сдвига и 3 сложения, используя равенство $W_2^{-m_1-n_2} + w_2^{-m_1-n_2} = (W_2 + w_2)^{-m_1-n_2}$.
4. Правую часть выражения для w_1 можно вычислить за 3 сдвига и 2 сложения, используя уже найденную сумму $W_2 + w_2$ и равенство $W_2^{-n_2} + w_2^{-n_2} = (W_2 + w_2)^{-n_2}$.

Итого: 47 сдвигов и 45 сложения.

Количество операций можно сократить. Одна из возможностей — преобразование f^{-1}. Например, сумму $x^4 + x^5 + x^8 + x^9$ его мономов, которая описывает 4 сдвига и 3 сложения, можно записать как $(x^4+x^5)+(x^4+x^5)^4$, сократив число операций до 3 сдвигов и 2 сложений.

Дальнейшую оптимизацию вычислений мы оставляем читателям. □