Методы защиты информации

А. Э. Маевский

1 Методы экспоненцирования

1.1 Экспоненцирование с фиксированной экспонентой

Определение 1. *Аддитивной цепочкой, вычисляющей число* n, называются две последовательности, $v = (v_0, \dots, v_s)$ — натуральных чисел и $w = (w_1, \dots, w_s)$ — пар натуральных чисел , таких что:

$$v_0 = 1,\ v_s = n$$
$$\forall i \in [1, s]\colon v_i = v_j + v_k, \quad 0 \leqslant j, k \leqslant i - 1;$$
$$w_i = (j, k).$$

Определение 2. А.Ц. называется *звёздной*, если $j = i - 1$ для всех i.

Пример 1.

$$v = (1, 2, 3, 6, 7, 14, 15)$$
$$w = ((0, 0), (0, 1), (22), (0, 3), (4, 4), (0, 5)).$$

Алгоритм 1.
Вход: $x \in G$, n, v, w — А.Ц. для n.
Выход: x^n.

(1) $x_1 := x$;

(2) Для i от 1 до s вычислить:

$$x_i := x_j \cdot x_k, \quad (j, k) = w_i.$$

(3) Вернуть x.

Определение 3. *Аддитивной-разностной цепочкой, вычисляющей число* n называется пара последовательностей, удовлетворяющая аналогичным А.Ц. требованиям, за исключением того, что для v_i требуется выполнение равенства $v_i = v_j + v_k$ либо $v_i = v_j - v_k$ для некоторых j, k.

Определение 4. Пусть $k, s \in \mathbb{N}$. *Векторной аддитивной цепочкой, вычисляющей последовательность* $(n_0, \ldots, n_{k-1})$, называются последовательности v — k-мерных векторов неотр. целых чисел и w — последовательность пар неотрицательных целых, таких что:

$$\begin{aligned} v_{-k+1} &= (1, 0, \ldots, 0), \\ v_{-k+2} &= (0, 1, \ldots, 0), \\ &\ldots \\ v_0 &= (0, 0, \ldots, 1), \end{aligned}$$

$$\begin{aligned} v_i = v_j + v_k, &\quad (j, k) = w_i, \\ v_s = (n_0 \ldots, n_{k-1}). \end{aligned}$$

Определение 5. *Словарём* $D = \{d_i\}$ *для числа* n называется набор целых чисел, такой что:

$$n = \sum_{i=0}^{k} b_{i,d_i} d_i 2^i, \quad b_{i,d_i} \in \{0, 1\},\ d_i \in D.$$

Упражнение 1. Убедиться, что словарём для 2^k-ичного метода явл $D = 1, 3, 5, \ldots, 2^k - 1$, а для знаковой модификации: $D = \{\pm 1, \pm 3, \ldots, \pm(2^k - 1)\}$.

1.1.1 Метод Кунихиро—Ямамото

Идея похожа на кодирование по Хаффману.

$$\begin{gathered} n = (n_{l-1}, \ldots, n_0)_2, \\ p = \frac{\text{кол-во «0»}}{l}, \\ q = 1 - p, \quad \hat{q} = \frac{1-p}{2}. \end{gathered}$$

Выбираем k — количество листов будущего дерева минус 1. Генерируем корень дерева. Затем потомки генерируются рекурсивно: от узла с весом w порождаются два потомка с весами в вершинах wp, wq и метками на дугах 0 и 1, соотв, для беззнакового представления и три потомка с весами wp, $w\hat{q}$, $w\hat{q}$ и метками на дугах 0, $\bar{1}0$, 10. Процедура повторяется для потомка с наибольшим весом.

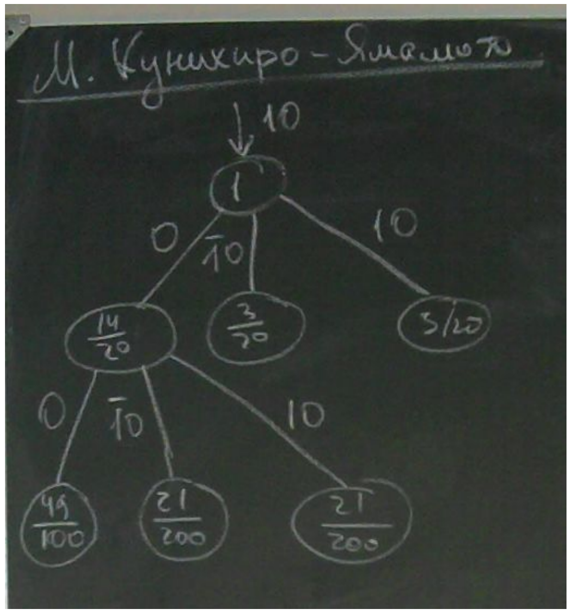
...

Проходим по всем маршрутам от корня к листьям, собирая метки на дугах. Удаляем trailing zeroes. Получаем словарь.

Пример 2. $n = 587257$, $k = 4$.

$$n = (10010000\bar{1}0\bar{1}00000\bar{1}001).$$

$$p = \frac{14}{20}. \quad \hat{q} = \frac{3}{20}.$$



$$
\begin{gathered}
1000 \Rightarrow 1 \Rightarrow 1,\\
100\bar{1}0 \Rightarrow 100\bar{1} \Rightarrow 7,\\
10010 \Rightarrow 1001 \Rightarrow 9,\\
10\bar{1}0 \Rightarrow 10\bar{1} \Rightarrow 3,\\
1010 \Rightarrow 101 \Rightarrow 5.
\end{gathered}
$$

$$D = \{1, 3, 5, 7, 9\}.$$

$$n = 9 \cdot 2^16 - 5 \cdot 2^9 - 7.$$

1.1.2 Метод Якоби

Кроме Хаффмана есть другие хорошие методы сжатия, например, метод Лемпела—Зива.

$n = (n_{l-1}, \ldots, n_0)$.

Пример 3. Просматривая двоичное представление справа-налево, выделяем новые куски и добавляем их в словарь.

$$
\begin{gathered}
n = 587257 = (\overline{10}\ \overline{00}\ \overline{1111}\ \overline{010}\ \overline{111}\ \overline{11}\ \overline{10}\ \overline{0}\ \overline{1})_2,\\
\{1, 0, 10, 11, 111, 010, 1111, 00\},\\
\{1, 0, 2, 3, 7, 2, 15, 0\}, \quad (\text{0 и двойки можно отбросить})\\
D = \{1, 3, 7, 15\}.
\end{gathered}
$$

Теперь можно получить разложение n по этому словарю:

$$n = 1 + 2^3 + 3 \cdot 2^4 + 7 \cdot 2^6 + 1 \cdot 2^1 1 + 15 \cdot 2^1 3 + 1 \cdot 20.$$

1.2 Экспоненцирование с фиксированным показателем

Первая идея: хранить «все» степени x.

1.2.1 Метод Яо

$$n = \sum_{i=0}^{l-1} n_i b_i, \quad n_i < h,$$

где h — некоторое фиксированное число.

$$x^n = \prod_{i=0}^{l-1} x_i^{n_i} = \prod_{j=1}^{h-1} \left(\prod_{i, n_i = j} x_i \right)^j$$

Пример 4. $x_i = x^{b_i}$, $i \in [0, l-1]$. x^{2989} — ? $h = 4$, $b_i = 4^i$.

$$2989 = (232231)_4.$$

$$x^{2989} = x_0^1 x_2^3 x_3^2 x_3^2 x_4^3 x_5^2 = (x_0)^1 (x_2 x_3 x_5)^2 (x_1 x_4)^3 = (x_1 x_4)(x_1 x_4 x_2 x_3 x_5)(x_1 x_4 x_2 x_3 x_5 x_0).$$

Алгоритм 2.

Вход:
Выход:

(1) $y := 1$, $u := 1$, $j := h - 1$.

(2) Пока $j \leqslant 1$ выполнять...

1.2.2 Метод, основанный на алгоритме Евклида

Алгоритм 3.

Вход: $x \in G$, $n = \sum n_i b_i$, $x_0 = x^{b_0}, \ldots, x_{l-1} = x^{b_{l-1}}$.
Выход:

(1) Повторять:

   (a) Найти M: $n_M \leqslant n_i$, $i \in [0, l-1]$.

(b) Найти N ($N \neq M$): $n_M \leqslant n_i$, $i \in [0, l-1] \setminus \{M\}$.

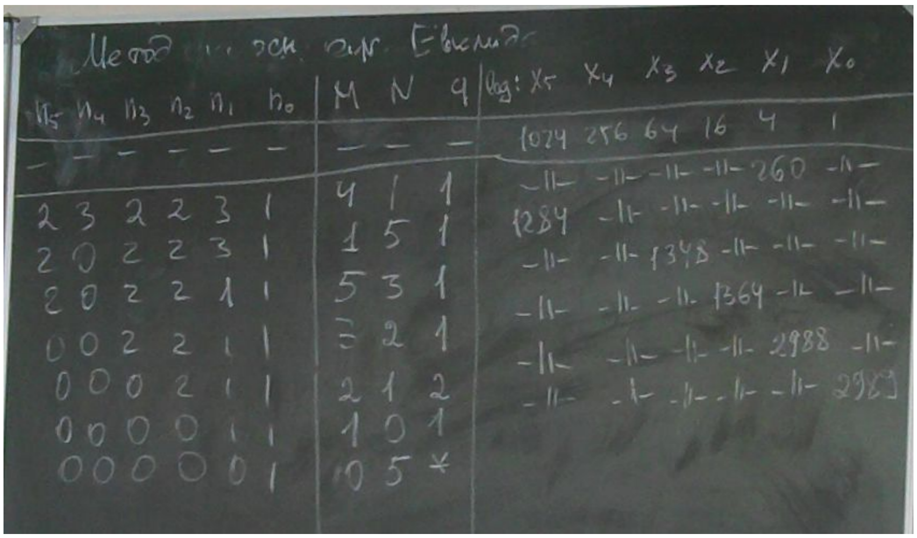
(c) Если $n_N \neq 0$, то

$$q := \lfloor n_M / n_N \rfloor, \quad x_N := x_M^q x_N, \quad n_M := n_M \mod n_N,$$

иначе прервать цикл.

(2) Вернуть x_M.

Пример 5. $x_i = x^{b_i}$, $i \in [0, l-1]$. x^{2989} — ? $h = 4$, $b_i = 4^i$, $2989 = (232231)_4$.



1.2.3 Метод Пайпенджера (Лима—Ли)

$n = (n_{l-1} \dots n_0)_2$, $h \in [1, l]_{\mathbb{N}}$, $\lceil l/h \rceil$.

$$\begin{array}{cccc}
a-1 & \dots & 1 & 0 \\
- & - & - & - \\
n_{a-1} & \dots & n_1 & n_0 \\
\dots & \dots & \dots & \dots \\
n_{ah-1} & \dots & n_{ah-a+1} & n_{ah-a}
\end{array}$$

$$G[j, i] = \left(\prod x^{i_s 2^{as}} \right)^{2^{jr}} \qquad j \in [0, v-1], \; i = (i_{h-1} \dots i_0) \in [0, 2^n - 1].$$

$$x^n = \prod_{k=0}^{r-1} \left(\prod_{j=0}^{v-1} G[j, I(jr+k)] \right)^{2^k}$$

$$I(s) = (n_{a(h-1)+s} \dots n_{a+s} n_s), \quad n = \dots$$

Алгоритм 4.

Вход: $x \in G$, $n \in \mathbb{N}$, $h, a, v, r, G[j,i]$.
Выход: $x^n \in G$.

(1) $y := 1$.

(2) Для $k := r - 1$ по 0 выполнить:

(a) $y := y^2$

(b) Для $j := v - 1$ по 0 выполнить:

$$I := \sum_{s=0}^{h-1} n_{as+jr+k} 2^s$$

(c) $y := y \cdot G[j, I]$

(3) Вернуть y.

Замечание (о сложности алгоритма). $a + r - 2$ умножений и $v(2^h - 1)$ сохранённых значений. Если возведение в квадрат быстрое, $v = 1$.