ICE

Криптоалгоритм ICE ($Information\ Concealment\ Engine$ – информационная шифровальная машина)[1] шифрует 64-битовые блоки открытых данных под управлением секретного ключа такого же размера.

Шифр построен в соответствии с 16-раундовой схемой Фейстеля (см. рис. 1). Раундовая функция F является композицией четырёх функций:

$$F(R, K_0, K_1, K_2) = P(S(KP(E(R), K_0, K_1, K_2))$$

Функция (перестановка с расширением) $E(R)$ от 32-битного аргумента $R = r_{31}r_{30} \dots r_0$, ($r_{31}$ – старший бит) возвращает 40-битовое значение $X = x_{39}x_{38} \dots x_0$:

$x_{39} \dots x_{30} = r_1r_0r_{31}r_{30}r_{29}r_{28}r_{27}r_{26}r_{25}r_{24}$,

$x_{29} \dots x_{20} = r_{25}r_{24}r_{23}r_{22}r_{21}r_{20}r_{19}r_{18}r_{17}r_{16}$,

$x_{19} \dots x_{10} = r_{17}r_{16}r_{15}r_{14}r_{13}r_{12}r_{11}r_{10}r_9r_8$,

$x_9 \dots x_0 = r_9r_8r_7r_6r_5r_4r_3r_2r_1r_0$.

Функция (ключевое преобразование) $KP(X, K_0, K_1, K_2)$ от 40-битового аргумента X и 20-битовых аргументов K_0, K_1, K_2 возвращает 40-битовое значение Y, вычисляемое следующим образом: пусть XL и YL обозначают левые (старшие), а XR и YR – правые (младшие) половины блоков X и Y; тогда:

$YL := K_2 \& (XL \oplus XR)$;

$YR := YL \oplus XR \oplus K_1$;

$YL := YL \oplus XL \oplus K_0$.

Функция (подстановка) $S(Y)$ от 40-битового аргумента $Y = y_{39}y_{38} \dots y_0$ возвращает 32-битовое значение $Z = z_{31}z_{30} \dots z_0$. Чтобы описать способ вычисления Z, представим блок Y в виде четырех 10-битовых подблоков:

$$A_0 = y_9 \dots y_0,\ A_1 = y_{19} \dots y_{10},\ A_2 = y_{29} \dots y_{20},\ A_3 = y_{39} \dots y_{30},$$

а блок Z – в виде четырех байтов:

$$B_0 = z_7 \dots z_0,\ B_1 = z_{15} \dots z_8,\ B_2 = z_{23} \dots z_{16},\ B_3 = z_{31} \dots z_{24}.$$

Биты a_9 и a_0 (старший и младший) в A_i определяют число $j = a_9 \times 2 + a_0 \in \{0,1,2,3\}$, а биты $a_8, a_7, \dots, a_1$ образуют некоторый байт C. Значение B_i вычисляется как:

$$U := C \oplus O_{ij};\ B_i := U^7.$$

Здесь O_{ij} – байт, значение которого определяется согласно табл. 1, а при вычислении значения U^7 байт U интерпретируется как элемент конечного поля $\mathbb{F}_{256} \cong \mathbb{F}_2[x]/f(x)$, где $f(x) = x^8 + g_{ij}(x)$, $g_{ij}(x)$ – многочлен 7-й степени, коэффициенты которого определяются согласно табл. 2.

Функция (перестановка) $P(z)$ от 32-битового аргумента Z возвращает 32-битовый блок, являющийся перестановкой битов блока Z. Перестановка P задана табл. 3.

Таблица 1

Значения битов O_{ij} в 16-ричном представлении

	0	1	2	3
0	ca	cd	$2e$	04
1	$4b$	$2e$	$4d$	33
2	cc	$a7$	ad	41
3	83	85	$9b$	cd

Таблица 2

Наборы коэффициентов многочленов $g_{ij}(x)$ в 16-ричном представлении

	0	1	2	3
0	$8d$	$a9$	$8b$	$f9$
1	69	bd	$c3$	$8d$
2	$7b$	77	$3f$	87
3	$4d$	39	$f9$	71

[1] Автор шифра: *Matthew Kwan* (Австралия)

Пояснение к табл. 2. Например, $8d$ определяет двоичный набор 10001101 и, соответственно, многочлен $x^7 + x^3 + x^2 + 1$.

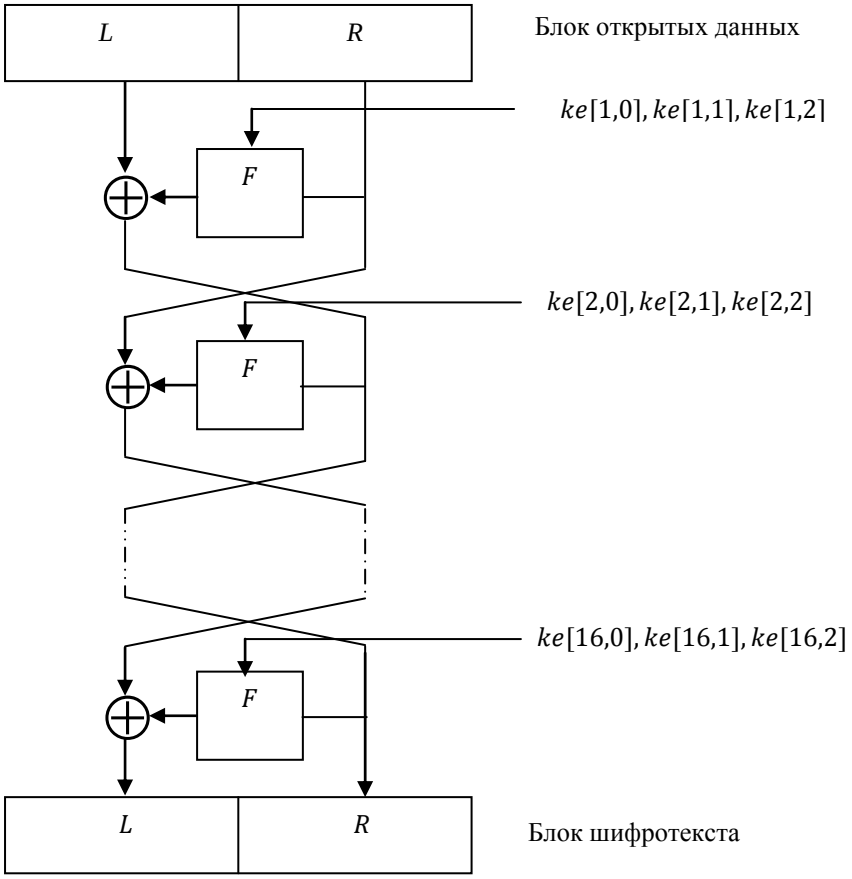

Рис. 1. *Структура* 16 *раундов шифра ICE*

Таблица 3

Перестановка P в ICE

31	7	15	23	22	14	30	6	13	21	5	29	4	28	20	12
19	11	3	27	26	2	18	10	1	25	9	17	8	16	24	0

Пояснение к табл. 3. Бит 31 остается на месте, бит 7 перемещается в позицию 30, …12 – в 16, 19 – в 15,…, 24 – в 1, бит 0 остается на месте.

В алгоритме зашифрования используются 20-битовые раундовые подключи $ke[n, j]$, $1 \le n \le 16$, $0 \le j \le 2$, генерируемые на этапе предвычислений на основе 64-битового секретного ключа $K = k_{63}k_{62} \ldots k_0$ (k_{63} – старший бит в K), причем в n-ом раунде используются подключи $K_1 = ke[n, 1]$, $K_2 = ke[n, 2]$ и $K_3 = ke[n, 3]$, образующие в совокупности 60-битовый подключ. Пусть b – битовая переменная (т.е. со значениями 0 и 1); $KB[0]$,…, $KB[3]$ – вспомогательные 20-битовые переменные. (*Замечание*: все переменные, а также раундовые подключи удобно хранить в 32-битовых словах, в которых 20-битовые подблоки занимают младшие биты 0,1,…19).

Пусть $bit_0(X)$ и $bit_{19}(X)$ обозначают соответственно младший и старший биты в 20-битовом блоке X. Массив $KR[1 \ldots 16]$ целых чисел определяется как:

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$KR[n]$	0	1	2	3	2	1	3	0	1	3	2	0	3	1	0	2

Алгоритм генерации раундовых подключей

Старт с нулевых значений $ke[n, i]$ и $KB[i]$;
$KB[0] := k_{63} \ldots k_{48}$;

(т.е. 16 старших битов ключа K помещаются в $KB[0]$ в качестве младших битов; 16 следующих – в $KB[1]$ и т.д.)

$KB[1] := k_{47} \dots k_{32}$;
$KB[2] := k_{31} \dots k_{16}$;
$KB[3] := k_{15} \dots k_{0}$;
for $n := 1$ ***to*** 16 ***do*** {
  for $j := 1$ ***to*** 3 ***do*** {
    for $t := 1$ ***to*** 5 ***do*** {
      for $i := 0$ ***to*** 3 ***do*** {
        $s := (i + KR[n]) \ mod \ 4$;
        $b := bit_0(KB[s])$;
        $shl_1(ke[n, j])$;
        $bit_0(ke[n, j]) := b$;
        $shr_1 \ (KB[s])$;
        $bit_{19} \ (KB[s]) := not \ b$
      }
    }
  }
}.

Алгоритм зашифрования ***ICE***

Вход: M – 64-битовый блок открытых данных в виде конкатенации 32-битовых подблоков L и R.

for $n := 1$ ***to*** 15 ***do*** {
  $L \coloneqq L \oplus F(R, ke[n, 0], ke[n, 1], ke[n, 2])$;
  $L \leftrightarrow R$
};
$L := L \oplus F(R, ke[16,0], ke[16,1], ke[16,2])$.
Выход: M – 64-битовый блок шифртекста.

Этот же алгоритм используется для расшифрования, но последовательность раундовых подключей $ke[n, i]$ заменяется на $kd[n, i]$:

$kd[n, i] := ke[17 - n, i], 1 \le n \le 16, 0 \le i \le 2$.