IDEA

Криптоалгоритм IDEA[1] (International Data Encryption Algorithm) шифрует 64-битовые блоки открытых данных под управлением 128-битового секретного ключа.

$IDEA$ не является шифром Фейстеля, но это симметричный шифр: алгоритм расшифрования идентичен алгоритму зашифрования (после определенного преобразования раундовых подключей). Общая схема алгоритма, состоящего из 8 раундов шифрования, представлена на рис.1. Основная идея конструкции – суррогатные вычисления (т.е. смешивание операций различных алгебраических групп).

В алгоритме используются операции над 16-битовыми подблоками (неотрицательными целыми числами) A и B:

$A \oplus B$ – побитовое сложение по модулю 2;
$A \boxplus B$ – сложение по модулю 2^{16};
$A \bullet B$ – умножение по модулю $2^{16}+1$.

Замечание. При выполнении операции умножения нулевой блок отождествляется с числом 2^{16}. Таким образом, операция умножения – это умножение в мультипликативной группе $\mathbb{Z}^*_{2^{16}+1} = \{1,2,\dots,2^{16}\}$ целых чисел по модулю $2^{16}+1$, а операция сложения – это сложение в аддитивной группе $\mathbb{Z}_{2^{16}} = \{0,1,\dots,2^{16}-1\}$ целых чисел по модулю 2^{16}. Обозначим через $\bar{X}$ элемент, обратный к X относительно сложения по модулю 2^{16}, а через Y^{-1} – элемент, обратный к Y относительно умножения по модулю $2^{16}+1$:

$$X \boxplus \bar{X} = 0, \qquad Y \bullet Y^{-1} = 1.$$

Отметим, что

$$\bar{X} = \begin{cases} 0, & \text{если X} = 0; \\ \text{0xFFFF} - (\text{X} - 1), & \text{если X} \neq 0, \end{cases}$$

где 0xFFFF – 16-ичная запись числа $2^{16}-1$, или $\bar{X} = \text{not}\,(\text{X}-1)$, где $\text{not}\,a$ – побитовое отрицание a. Значение Y^{-1} можно вычислить, используя следующее соотношение, вытекающее из Малой теоремы Ферма:

$$Y^{-1} = Y^{2^{16}-1}\ mod\ 2^{16}+1.$$

Отметим также, что $a \bullet 0 = 0 \bullet a = \text{not}\,(a+2)\ mod\ 2^{16}$; если $a, b > 0$, то

$$a \bullet b = \begin{cases} (ab\ mod\ 2^{16}) - (ab\ div\ 2^{16}), & \text{если } ab\ mod\ 2^{16} \geq ab\ div\ 2^{16}; \\ (ab\ mod\ 2^{16}) + 2^{16} + 1 - (ab\ div\ 2^{16}), & \text{если } ab\ mod\ 2^{16} < ab\ div\ 2^{16}. \end{cases}$$

В алгоритме используются следующие преобразования над 64-битовым блоком $\text{M} = (M_1, M_2, M_3, M_4)$, представленным в виде четырех 16-битовых подблоков M_1, M_2, M_3 и M_4, под управлением 16-битовых ключей $Q_1, \dots, Q_6$ (далее $X_1, X_2, Y_1, Y_2, Z_1, Z_2$ – вспомогательные 16-битовые переменные):

$\mathcal{E}_1(M, Q_1, Q_2, Q_3, Q_4) \equiv \{M := (M_1 \bullet Q_1, M_2 \boxplus Q_2, M_3 \boxplus Q_3, M_4 \bullet Q_4)\};$
$\mathcal{E}_2(M, Q_5, Q_6) \equiv \{$
$X_1 := M_1 \oplus M_3;$
$X_2 := M_2 \oplus M_4;$
$Y_1 := X_1 \bullet Q_5;$
$Y_2 := X_2 \boxplus X_1 \bullet Q_5);$
$Z_1 := Y_2 \bullet Q_6;$
$Z_2 := Y_1 \boxplus Y_2 \bullet Q_6);$
$M := M \oplus (Z_1, Z_2, Z_1, Z_2);$
$M_2 \leftrightarrow M_3$
$\},$

где запись $M_2 \leftrightarrow M_3$ означает, что M_2 и M_3 обмениваются значениями.

[1] Авторы шифра: *Xuejia Lai, James Massey* (Швейцария, Федеральный институт технологий ETH Zurich, 1990)

Обратные преобразования, возвращающие блок M к исходному значению, имеют следующий вид:

$\mathcal{E}_1^{-1}(M, Q_1, Q_2, Q_3, Q_4) \equiv \mathcal{E}_1(M, Q_1^{-1}, \overline{Q}_2, \overline{Q}_3, Q_4^{-1})$;

$\mathcal{E}_2^{-1}(M, Q_5, Q_6) \equiv \{$

$X_1 := M_1 \oplus M_2$;

$X_2 := M_3 \oplus M_4$;

$Y_1 := X_1 \bullet Q_5$;

$Y_2 := X_2 + (X_1 \bullet Q_5)$;

$Z_1 := Y_2 \bullet Q_6$;

$Z_2 := Y_1 \boxplus (Y_2 \bullet Q_6)$;

$M_2 \leftrightarrow M_3$;

$M := M \oplus (Z_1, Z_2, Z_1, Z_2)$

$\}$.

Отметим, что $\mathcal{E}_2^{-1}(M, Q_5, Q_6) \equiv \{M_2 \leftrightarrow M_3;\ \mathcal{E}_2(M, Q_5, Q_6); M_2 \leftrightarrow M_3\}$.

В алгоритме зашифрования (расшифрования) используются 52 раундовых подключей $k_1, \ldots, k_{52}$, формируемых на основе 128-битового секретного ключа K: k_1 равен первым (наиболее значимым) 16 битам ключа K, k_2 – следующим 16 битам, k_8 – последним 16 битам; затем ключ K циклически сдвигается влево на 25 битов и создаются восемь следующих подключей – $k_9, \ldots, k_{16}$. Эта процедура повторяется, пока не будут получены все 52 подключа.

Алгоритм зашифрования

Вход: P – 64-битовый блок открытых данных в виде четырех 16-битовых подблоков P_1, P_2, P_3, P_4.

1. (8 раундов зашифрования.)

for $i := 0$ ***to*** 7 ***do*** $\{$

$\mathcal{E}_1(P, k_{6i+1}, k_{6i+2}, k_{6i+3}, k_{6i+4})$;

$\mathcal{E}_2(P, k_{6i+5}, k_{6i+6})$

$\}$;

2. (Выходное преобразование.)

$P_2 \leftrightarrow P_3$;

$\mathcal{E}_1(P, k_{49}, k_{50}, k_{51}, k_{52})$.

Выход: P – 64-битовый блок шифртекста.

Таблица 1

Расписание использования раундовых подключей в IDEA

Раунд	Зашифрование	Расшифрование
1	$k_1 k_2 k_3 k_4 k_5 k_6$	$k_{49}^{-1} \overline{k}_{50} \overline{k}_{51} k_{52}^{-1} k_{47} k_{48}$
2	$k_7 k_8 k_9 k_{10} k_{11} k_{12}$	$k_{43}^{-1} \overline{k}_{45} \overline{k}_{44} k_{46}^{-1} k_{41} k_{42}$
3	$k_{13} k_{14} k_{15} k_{16} k_{17} k_{18}$	$k_{37}^{-1} \overline{k}_{39} \overline{k}_{38} k_{40}^{-1} k_{35} k_{36}$
4	$k_{19} k_{20} k_{21} k_{22} k_{23} k_{24}$	$k_{31}^{-1} \overline{k}_{33} \overline{k}_{32} k_{34}^{-1} k_{29} k_{30}$
5	$k_{25} k_{26} k_{27} k_{28} k_{29} k_{30}$	$k_{25}^{-1} \overline{k}_{27} \overline{k}_{26} k_{28}^{-1} k_{23} k_{24}$
6	$k_{31} k_{32} k_{33} k_{34} k_{35} k_{36}$	$k_{19}^{-1} \overline{k}_{21} \overline{k}_{20} k_{22}^{-1} k_{17} k_{18}$
7	$k_{37} k_{38} k_{39} k_{40} k_{41} k_{42}$	$k_{13}^{-1} \overline{k}_{15} \overline{k}_{14} k_{16}^{-1} k_{11} k_{12}$
8	$k_{43} k_{44} k_{45} k_{46} k_{47} k_{48}$	$k_7^{-1} \overline{k}_9 \overline{k}_8 k_{10}^{-1} k_5 k_6$
Выходное преобразование	$k_{49} k_{50} k_{51} k_{52}$	$k_1^{-1} \overline{k}_2 \overline{k}_3 k_4^{-1}$

Алгоритм расшифрования

При расшифровании обратные преобразования выполняются в обратном порядке, а именно:

$\mathcal{E}_1(P,\ k_{49}^{-1}, \overline{k}_{50}, \overline{k}_{51}, k_{52}^{-1});\ P_2 \leftrightarrow P_3;$
$\boldsymbol{for}\ i := 7\ \boldsymbol{downto}\ 0\ \boldsymbol{do}\ \{$
$P_2 \leftrightarrow P_3;$
$\mathcal{E}_2(P, k_{6i+5}, k_{6i+6});$
$P_2 \leftrightarrow P_3;$
$\mathcal{E}_1(P, k_{6i+1}^{-1}, \overline{k}_{6i+2}, \overline{k}_{6i+3}, k_{6i+4}^{-1},)$
$\}$.

Поскольку результаты применения к блоку P преобразований

$$\{P_2 \leftrightarrow P_3;\ \mathcal{E}_1(P, Q_1, Q_2, Q_3, Q_4);\ P_2 \leftrightarrow P_3\}$$

и

$$\mathcal{E}_1(P, Q_1, Q_2, Q_3, Q_4)$$

совпадают, то алгоритм расшифрования приводится к виду:

$\mathcal{E}_1(P, k_{49}^{-1}, \overline{k}_{50}, \overline{k}_{51}, k_{52}^{-1});$
$\mathcal{E}_2(P, k_{47}, k_{48});$
$\boldsymbol{for}\ i := 0\ \boldsymbol{to}\ 6\ \boldsymbol{do}\ \{$
$\mathcal{E}_1(P, k_{43-6i}^{-1}, \overline{k}_{45-6i}, \overline{k}_{44-6i}, k_{46-6i}^{-1});$
$\mathcal{E}_2(P, k_{47-6i}, k_{48-6i})$
$\};$
$P_2 \leftrightarrow P_3;$
$\mathcal{E}_1(P, k_1^{-1}, \overline{k}_2, \overline{k}_3, k_4^{-1})$.

Сравнивая алгоритмы, нетрудно убедиться в том, что для зашифрования и расшифрования может быть использован один и тот же алгоритм. Различие в использовании раундовых подключей отражено в таблице 1.

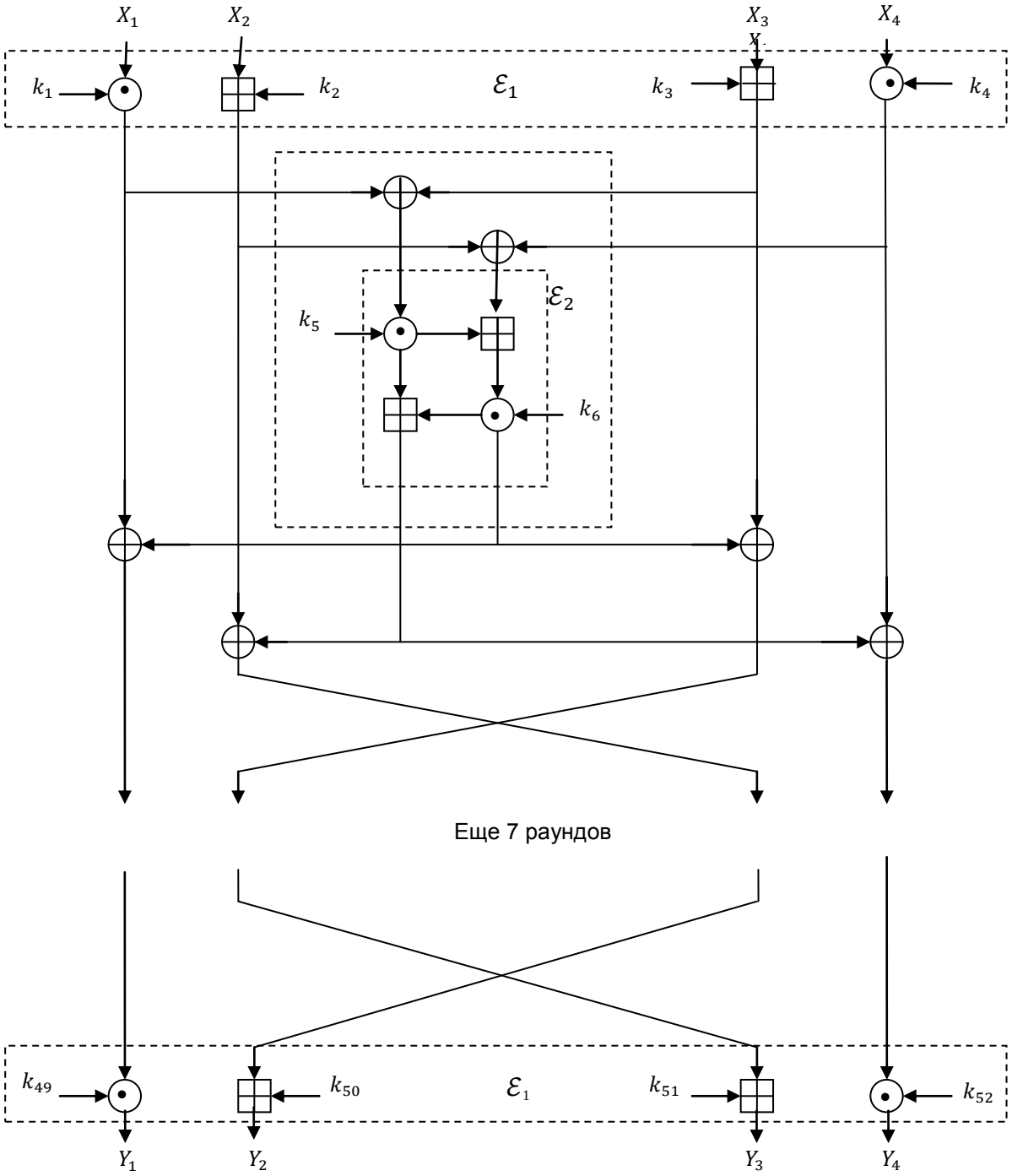

Рис. 1. *Структура алгоритма IDEA*