Crypton V1.0

Криптоалгоритм $Crypton\ V1.0$ [1] шифрует 128-битовые блоки открытых данных под управлением 256-битового 16-байтовые блоки $B = (b_0 b_1 \dots b_{15})$, участвующие в криптографическом преобразовании, представляются в виде 4×4-матрицы

$$\mathrm{A} = (a_{ij}) = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} b_0 & b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 & b_7 \\ b_8 & b_9 & b_{10} & b_{11} \\ b_{12} & b_{13} & b_{14} & b_{15} \end{pmatrix}.$$

В алгоритме используются следующие преобразования над 4×4-матрицами:

1) Нелинейные преобразования γ_n, $n = 0$ или 1 определяются как

$$\gamma_n((a_{ij})) = (b_{ij});\ \ b_{ij} = S_{i+j+2n \bmod 4}[a_{ij}],\ 0 \le i, j \le 3.$$

Здесь S_0, S_1, S_2 и S_3 – подстановки на множестве байтов, причем $S_2 = S_0^{-1}$, $S_3 = S_1^{-1}$, (ввиду этого $\gamma_0^{-1} = \gamma_1$ и $\gamma_1^{-1} = \gamma_0$). Подстановки S_0 и S_1 конструируются следующим образом. Сначала на основе двух подстановок P_0 и P_1, заданных на множестве полубайтов (см. в табл. 1), строится инволютивная подстановка $y = S[x]$, заданная на множестве байтов (далее x_7, x_6, …, x_0 – биты, образующие байт x со значением $x_7 2^7 + x_6 2^6 + \dots + x_0$; полубайты $x_7x_6x_5x_4$ и $x_3x_2x_1x_0$ имеют значения $x_7 2^3 + x_6 2^2 + x_5 2 + x_4$ и $x_3 2^3 + x_2 2^2 + x_1 2 + x_0$; аналогично, w_i, y_i и z_i – биты, образующие байты w, y и z:

$$z_7z_6z_5z_4 := P_1[x_7x_6x_5x_4];\ \ z_3z_2z_1z_0 := P_0[x_3x_2x_1x_0];$$

$$\begin{pmatrix} w_7 \\ w_6 \\ w_5 \\ w_4 \\ w_3 \\ w_2 \\ w_1 \\ w_0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} z_7 \\ z_6 \\ z_5 \\ z_4 \\ z_3 \\ z_2 \\ z_1 \\ z_0 \end{pmatrix};$$

$$y_7y_6y_5y_4 := P_1^{-1}[w_7w_6w_5w_4];$$
$$y_3y_2y_1y_0 := P_0^{-1}[w_3w_2w_1w_0];$$
$$S[x_7x_6x_5x_4x_3x_2x_1x_0] := y_7y_6y_5y_4y_3y_2y_1y_0.$$

На основе подстановки S определяются подстановки S_i:

$\boldsymbol{for}\ x := 0\ to\ 255\ \boldsymbol{do}$

{

$S_0[x] := rol_1(S[x]);\ \ S_1[x] := rol_3(S[x]);$

$S_2[x] := S[rol_7(x)];\ \ S_3[x] := S[rol_5(x)]$

},

где $rol_m(y)$ – циклический сдвиг байта y влево на m битовых позиций.

Таблица 1

Подстановки P_0 и P_1

	0	1	2	3	4	5	6	7	8	9	*a*	*b*	*c*	*d*	*e*	*f*
P_0	*f*	*e*	*a*	1	b	5	8	d	9	3	2	7	0	6	4	*c*
P_1	*b*	*a*	*d*	7	8	*e*	0	5	*f*	6	3	4	1	9	2	c
P_0^{-1}	*c*	3	*a*	9	e	5	*d*	*b*	6	8	2	4	*f*	7	1	0
P_1^{-1}	6	*c*	*e*	*a*	*b*	7	9	3	4	*d*	1	0	*f*	2	5	8

2) Преобразования π_n, $n = 0$ или 1, определяются как

[1] Автором шифра является *Chae Hoon Lim* (Южная Корея).

$$\pi_n((a_{ij})) = (b_{ij}),$$

где

$$b_{ij} = \bigoplus_{k=0}^{3} m_{i-j+3+k+2n \; mod \; 4} \& \, a_{kj}, \; 0 \le i, j \le 3;$$
$$m_0 = 0\text{x}fc, m_1 = 0xf3, m_2 = 0xcf, m_3 = 0x3f.$$

Отметим, что $\pi_n^{-1} = \pi_n$, $n = 1, 2$.

3) Преобразование $\tau((a_{ij})) = (a_{ji})$ − обычное транспонирование матрицы (a_{ij}). Очевидно, что $\tau^{-1} = \tau$.

Алгоритм зашифрования

Вход : P – 128-битовый (16-байтовый) блок открытых данных в виде 4×4-матрицы.

В алгоритме зашифрования используются раундовые подключи ke_0, ke_1,…, ke_{12}, представленные, как и блок P, в виде 4×4-матриц

$C := P;$
$C := C \oplus ke_0;$
$\boldsymbol{for} \; r := 1 \; \boldsymbol{to} \; 12 \; \boldsymbol{do}$ {
$\quad n := (r-1) \; mod \; 2;$
$\quad \tau(\pi_n(\gamma_n(C)));$
$\quad C := C \oplus ke_r$
};
$\tau(\pi_1(\tau(C)))$.

Выход: C – 128-битовый блок шифртекста.

Замечание. Можно показать, что алгоритм зашифрования пригоден и для расшифрования, если последовательность раундовых подключей ke_i заменить на kd_i:

$kd_i := \tau(\pi_{(\mathrm{i}+1) mod \; 2}(\tau(ke_{12-i})))$, $\; i := 0,1, \dots, 12$.

Алгоритм расшифрования

Вход: C – 128-битовый блок шифртекста в виде 4×4-матрицы.

$P := C;\quad \tau(\pi_1(\tau(P)));$
$\boldsymbol{for} \; r := 12 \; \boldsymbol{downto} \; 1 \; \boldsymbol{do}${
$\quad P := P \oplus ke_r;$
$\quad \gamma_{r \; mod \; 2}\left(\pi_{(r+1) \mathrm{mod} \; 2}\left(\tau(P)\right)\right)$
};
$P := P \oplus ke_0$.

Выход: P – 128-битовый блок открытых данных.

Вычисление раундовых подключей.

Раундовые подключи ke_0, ke_1,…, ke_{12} генерируются на основе 32-байтового секретного ключа $K = (k_0, k_1, \dots, k_{31})$. Каждый раундовый ключ представлен 4×4-матрицей; -ая строка матрицы ke_r обозначается $ke_{ri,}$ $0 \le i \le 3$, $0 \le r \le 12$. При вычислении значений ke_{ri} используются вспомогательные переменные u_0, u_1, u_2, u_3; v_0, v_1, v_2, v_3; $e_0, e_1, \dots, e_7$ и константы $\mathrm{c}_0, \mathrm{c}_1, \dots, \mathrm{c}_{12}$; mc_0, mc_1, mc_2, mc_3. Каждая из них рассматривается либо как 4-байтовый массив, либо как 32-битовое число. Операции rol_n и $rolb_n$ определяются следующим образом:

$rol_n(X)$ – циклический сдвиг 32-битового числа X влево на n битов;

$rolb_n(X)$ – циклический сдвиг каждого байта в 4-байтовом массиве X влево на n битов;

$$\begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{pmatrix} := \tau\left(\pi_0\left(\gamma_0\begin{pmatrix} k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k_{10} & k_{11} \\ k_{12} & k_{13} & k_{14} & k_{15} \end{pmatrix}\right)\right);$$

$$\begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} := \tau\left(\pi_1\left(\gamma_1\begin{pmatrix} k_7 & k_5 & k_2 & k_1 \\ k_{15} & k_{13} & k_{11} & k_9 \\ k_{23} & k_{21} & k_{19} & k_{17} \\ k_{31} & k_{29} & k_{27} & k_{25} \end{pmatrix}\right)\right);$$

$\boldsymbol{for}\ i := 0\ \boldsymbol{to}\ 3\ \boldsymbol{do}\ \{$

  $e_i := u_i \oplus v_0 \oplus v_1 \oplus v_2 \oplus v_3;$

  $e_{i+4} := v_i \oplus u_0 \oplus u_1 \oplus u_2 \oplus u_3$

$\};$

$c_0 := 0\text{x}a54ff53a;$

$\boldsymbol{for}\ i := 1\ \boldsymbol{to}\ 12\ \boldsymbol{do}\ \ c_i := (c_{i-1} + 0x3c6ef372)\ mod\ 2^{32};$

$mc_0 := 0\text{x}acacacac;$

$\boldsymbol{for}\ i := 0\ \boldsymbol{to}\ 3\ \boldsymbol{do}\ \ mc_i := rolb_1\,(mc_{\mathrm{i}-1});$

$\boldsymbol{for}\ i := 0\ \boldsymbol{to}\ 3\ \boldsymbol{do}\ \{$

  $ke_{0,i} := e_i \oplus c_0 \oplus mc_i;$

  $ke_{1,i} := e_{\mathrm{i}+4} \oplus c_1 \oplus mc_i$

$\};$

$\boldsymbol{for}\ r := +2\ \boldsymbol{to}\ 12\ \boldsymbol{do}\ \{$

  $\boldsymbol{if}\ r$ нечетно $\boldsymbol{then}$

  $\{$

    $(e_4, e_5, e_6, e_7) := (rolb_2(e_7), rolb_2(e_4), rol_8(e_5), rol_{16}(e_6));$

    $\boldsymbol{for}\ i := 0\ \boldsymbol{to}\ 3\ \boldsymbol{do}\ \ ke_{r,i} := e_{i+4} \oplus c_r \oplus mc_i$

  $\}$

  $\boldsymbol{else}$

  $\{$

    $(e_0, e_1, e_2, e_3) := (rol_{24}(e_1), rol_{16}(e_2), rolb_6(e_3), rolb_6(e_0));$

    $\boldsymbol{for}\ i := 0\ \boldsymbol{to}\ 3\ \boldsymbol{do}\ \ ke_{r,i} := e_i \oplus c_r \oplus mc_i;$

  $\}$

$\}.$