E2

Криптоалгоритм $E2$ [1] шифрует 128-битовые блоки открытых данных под управлением секретного ключа, длина которого может составлять 128, 196 или 256 битов.

По своей структуре шифр $E2$ – классический шифр Фейстеля. В алгоритме используются следующие преобразования:

$X \oplus Y$ – побитовое сложение по модулю 2 блоков X и У одинаковой длины (в 1, 8 или 16 байтов).

В операциях $X \otimes Y = U$ и $X \oslash Y = W$ аргументы и результат – 16-байтовые блоки, представленные в виде массивов 32-битовых слов:

$X = x_1x_2x_3x_4$, $Y = y_1y_2y_3y_4$, $U = u_1u_2u_3u_4$, $W = w_1w_2w_3w_4$.

Значения u_i и w_i вычисляются как

$u_i = x_i \cdot (y_i \vee 1)$, $w_i = x_i \cdot (y_i \vee 1)^{-1}$, $i = 1, 2, 3, 4$.

Слова x_i, y_i, u_i, w_i рассматриваются в данном случае как неотрицательные целые 32-разрядные числа (в которых левый байт считается старшим).

Операция умножения $(\cdot)$ выполняется по модулю 2^{32}. Значения $(y_i \vee 1)$ и z^{-1} определяются как

$$(y_i \vee 1) = \begin{cases} \text{у, если у – нечётное число;} \\ \text{у} + 1\text{, если у – чётное число;} \end{cases}$$

$z^{-1} = z^{2^{31}-1} mod\ 2^{32}$

(z^{-1} – число, обратное к z относительно умножения по модулю 2^{32}, т.е. $z \cdot z^{-1} mod\ 2^{32} = 1$, если z – нечётное число). Отметим, что $(X \otimes \text{У}) \oslash \text{У} \equiv X$.

Функция BP от 16-байтового аргумента $X = (x_0, x_1, \dots, x_{15})$ определяется как

$BP(X) = (x_0, x_5, x_{10}, x_{15}, x_4, x_9, x_{14}, x_3, x_8, x_{13}, x_2, x_7, x_{12}, x_1, x_6, x_{11})$.

Другими словами, BP – перестановка байтов в X: в позицию j перемещается байт из позиции $5j\ mod\ 16$, $j = 0,1,2,\dots,15$. Обратная функция BP^{-1} возвращает X к исходному значению: в позицию j перемещается байт из позиции $13j\ mod\ 16$, $j = 0,1,2,\dots,15$, т.е.

$BP^{-1}(\text{X}) = (x_0, x_{13}, x_{10}, x_7, x_4, x_1, x_{14}, x_{11}, x_8, x_5, x_2, x_{15}, x_{12}, x_9, x_6, x_3)$.

Функция BRL возвращает 8-байтовый аргумент $\text{X} = (x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7)$, циклически сдвинутый на один байт влево:

$BRL(X) = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_0)$.

Функция S от 8-байтового аргумента $\text{X} = (x_0, x_1, \dots, x_7)$ возвращает 8-байтовое значение $S(X) = (s(x_0), s(x_1), \dots, s(x_1))$, где s – подстановка на множестве байтов, являющаяся произведением двух подстановок:

$Power: x \to x^e$,

$Affine: x \to a \cdot x + b$.

В подстановке $Power$ байты интерпретируются как элементы конечного поля $\mathbb{F}_{256} = \mathbb{F}_2[x]/r(x)$, где $r(x) = x^8 + x^4 + x^3 + x + 1$ (соответственно и операция возведения в степень x^e выполняется в этом поле); в подстановке $Affine$ байты интерпретируются как целые числа (более точно: как элементы кольца $\mathbb{Z}_{256}$), а операции умножения и сложения выполняются по модулю 256. Значение $s(x)$ определяется как

$s(x) = Affine(Power(x, 127), 97, 225) = 97(x^{127}) + 225$.

Подстановка s также представлена в табл.1.

Функция $P(\text{X})$ от 8-байтового аргумента $\text{X} = (x_0, x_1, \dots, x_7)$ возвращает 8-байтовое значение $\text{У} = (y_0, y_1, \dots, y_7)$:

[1] Авторы шифра: сотрудники корпорации NTT и специалисты университета г. Иокагама (Япония)

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} := \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix},$$

т.е. $y_0 := x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6$ и т.д. Значения y_i можно последовательно вычислить как

$$y_7 := x_7 \oplus x_3;\ \ y_6 := x_6 \oplus x_2;\ \ y_5 := x_5 \oplus x_1;\ \ y_4 := x_4 \oplus x_0;$$
$$y_3 := x_3 \oplus x_5;\ \ y_2 := x_2 \oplus x_4;\ \ y_1 := x_1 \oplus x_7;\ \ y_0 := x_0 \oplus x_6;$$
$$y_7 := y_7 \oplus y_2;\ \ y_6 := y_6 \oplus y_1;\ \ y_5 := y_5 \oplus y_0;\ \ y_4 := y_4 \oplus y_3;$$
$$y_3 := y_3 \oplus y_7;\ \ y_7 := y_2 \oplus y_6;\ \ y_1 := y_1 \oplus y_5;\ \ y_0 := y_0 \oplus y_4.$$

Раундовая функция $F(R,k)$, зависящая от 8-байтового аргумента R и 16-байтового аргумента k, определяется как:

$$F(R,k) = BRL(S(P(S(R \oplus kl)) \oplus kr)),$$

где kl и kr – левая (старшая) и правая (младшая) половины k.

Вычисление раундовых подключей

В алгоритме $E2$ используются шестнадцать 16-байтовых раундовых подключей $k_1, k_2, \ldots, k_{16}$, формируемых на основе секретного ключа K с использованием функции $G(X,U)$ от 32-байтового аргумента X и 8-байтового аргумента U, возвращающей вектор $(L,\ Y,\ V)$ с 32-байтовыми компонентами L и Y и 8-байтовым компонентом V.

Представим X, L и Y в виде массивов 8-байтовых подблоков X_i, L_i и Y_i:

$$X = (X_0, X_1, X_2, X_3),\ L = (L_0, L_1, L_2, L_3),\ Y = (Y_0, Y_1, Y_2, Y_3);$$

Значения L, Y и V, возвращаемые функцией G, вычисляются следующим образом:

for $i := 0$ ***to*** 3 ***do*** $Y_i := P(S(X_i))$;
$L_0 := Y_0 \oplus P(S(U))$;
for $i := 0$ ***to*** 3 ***do*** $L_i := P(S(L_{i-1}))$;
$V := L_3$.

Предполагается, что секретный ключ K имеет длину в 256 битов и представлен в виде четырёх 8-байтовых подблоков, т.е. $K = (K_0, K_1, K_2, K_3)$. Если K – 128-битовый ключ, то его расширяют, полагая $K_3 = S(S(S(g)))$, $K_4 = S(S(S(S(g)))) = S(K_3)$, где $g = 0x0123456789abcdef$ – 8-байтовая константа (0x01 – левый байт); если K – 192-битовый ключ, то полагают $K_4 = S(S(S(S(g))))$.

В алгоритме формирования 16-байтовых раундовых подключей используется массив $(q_0, q_1, \ldots, q_{31})$, с 8-байтовыми компонентами $q_i = (q_{i0}, q_{i1}, \ldots, q_{i7})$:

$U := g$;
$(L, Y, U) := G(K, U)$;
$p := 0$;
for $i := 0$ ***to*** 7 ***do***
{
$(L, Y, U) := G(Y, U)$;
$(q_{4i}, q_{4i+1}, q_{4i+2}, q_{4i+3}) := L$
};
for $i := 0$ **to** 7 ***do*** {
$k_{2i+1} := (q_{0p}\ q_{2p}\ \ldots\ q_{30p})$;
$k_{2i+2} := (q_{1p}\ q_{3p}\ \ldots\ q_{31p})$;

$p := p + 1$

}.

Алгоритм зашифрования

Вход: $M = L||R$ – 128-битовый блок открытых данных, представленный в виде конкатенации 8-байтовых подблоков L и R.

1. (Начальное преобразование.)

$M := BP((M \oplus k_{13}) \otimes k_{14})$;

2. (12 раундов шифрования по схеме Фейстеля.)

for $i := 1$ ***to*** 11 ***do***

{

$L := L \oplus F(R, k_i)$;

$L \leftrightarrow R$

};

$L := L \oplus F(R, k_{12})$;

3. (Заключительное преобразование.)

$C := (BP^{-1}(M) \oslash k_{15}) \oplus k_{16}$.

Выход: $C = L||R$ – 128-битовый блок шифртекста.

Алгоритм расшифрования

Для расшифрования используется тот же алгоритм, что и для зашифрования, но последовательность раундовых подключей $k_1, k_2, \ldots, k_{12}, k_{13}, k_{14}, k_{15}, k_{16}$ преобразуется к виду: $k_{12}, k_{11}, \ldots, k_1, k_{16}, k_{15}, k_{14}, k_{13}$.

Таблица 1

Подстановка S в E2

225	66	62	129	78	23	158	253	180	63	44	218	49	30	224	65
204	243	130	125	124	18	142	187	228	88	21	213	111	233	76	75
53	123	90	154	144	69	188	248	121	214	27	136	2	171	207	100
9	12	240	1	164	176	246	147	67	99	134	220	17	165	131	139
201	208	25	149	106	161	92	36	110	80	33	128	47	231	83	15
145	34	4	237	166	72	73	103	236	247	192	57	206	242	45	190
93	28	227	135	7	13	122	244	251	50	245	140	219	143	37	150
168	234	205	51	101	84	6	141	137	10	94	217	22	14	113	108
11	255	96	210	46	211	200	85	194	35	183	116	226	155	223	119
43	185	60	98	19	229	148	52	177	39	132	159	215	81	0	97
173	133	115	3	8	64	239	104	254	151	31	222	175	102	232	184
174	189	179	235	198	107	71	169	216	167	114	238	29	126	170	182
117	203	212	48	105	32	127	55	91	157	120	163	241	118	250	5
61	58	68	87	59	202	199	138	24	70	156	191	186	56	86	26
146	77	38	41	162	152	16	153	112	160	197	40	193	109	20	172
249	95	79	196	195	209	252	221	178	89	230	181	54	82	74	42