RC

***RC2*.** Криптоалгоритм $RC2$ [1] шифрует 64-битовые блоки данных под управлением секретного ключа K, длина m которого может варьироваться от 1 до 128 байтов.

Ключ K сохраняется в массиве байтов $b_0, b_1,\ldots, b_{127}$. При описании операций шифрования удобнее ссылаться на ключ как на массив 16-битовых слов $k_0, k_1,\ldots, k_{63}$. Если длина ключа < 128 байтов, то в алгоритме предусмотрена следующая процедура расширения ключа до требуемого размера с помощью дополнительного массива псевдослучайных байтов $p_0, p_1,\ldots, p_{255}$, полученных с использованием десятичных знаков числа π:

for $i := m$ ***to*** 127 ***do*** {
    $j := (b_{i-1} + b_{i-m})\ mod\ 256$;
    $b_i := p_j$
};
$j := b_{128-m}$;
$b_{128-m} := p_{128-m}$;
for $i := 127 - m$ ***downto*** 0 ***do***{
    $j := b_{i+1} \oplus b_{i+m}$;
    $b_i := p_j$
}.

В алгоритме используются элементарные операции над 16-битовыми подблоками: сложение $(+)$ и вычитание $(-)$ по модулю 2^{16}, побитовое сложение по модулю 2 $(\oplus)$, конъюнкция $(\&)$, отрицание $(\neg)$, циклический сдвиг влево на s позиций (rol_s). Основу алгоритмов зашифрования составляют преобразования $Emixing$ (смешивание) и $Emashing$ (сплющивание), выполняемые над 64-битовым блоком M, представленным в виде конкатенации 16-битовых подблоков M_0, M_1, M_2 и M_3:

$Emixing(M, j) \equiv$ {
    $M_0 := M_0 + k_j \quad + (M_1 \& (\neg M_3)) + (M_2 \& M_3);\ rol_1(M_0);$
    $M_1 := M_1 + k_{j+1} + (M_2 \& (\neg M_0)) + (M_3 \& M_0);\ rol_2(M_1);$
    $M_2 := M_2 + k_{j+2} + (M_3 \& (\neg M_1)) + (M_0 \& M_1);\ rol_3(M_2);$
    $M_3 := M_3 + k_{j+3} + (M_0 \& (\neg M_2)) + (M_1 \& M_2);\ rol_5(M_3)$
}.

$Emashing(M) \equiv$ {
    $j := M_3 \& 63;\ M_0 := M_0 + k_j;$
    $j := M_0 \& 63;\ M_1 := M_1 + k_j;$
    $j := M_1 \& 63;\ M_2 := M_2 + k_j;$
    $j := M_2 \& 63;\ M_3 := M_3 + k_j$
}.

Обратные преобразования, возвращающие блок M к исходному значению, задаются следующим образом:

$Dmixing(M, j) \equiv$ {
    $rol_{11}(M_3);\ M_3 := M_3 - (k_{j+3} + (M0 \& (\neg M_2)) + (M_1 \& M_2));$
    $rol_{13}(M_2);\ M_2 := M_2 - (k_{j+2} + (M_3 \& (\neg M_1)) + (M_0 \& M_1));$
    $rol_{14}(M_1);\ M_1 := M_1 - (k_{j+1} + (M_2 \& (\neg M_0)) + (M_3 \& M_0));$
    $rol_{15}(M_0);\ M_0 := M_0 - (k_j \quad + (M_1 \& (\neg M3)) + (M_2 \& M_3))$
}.

$Dmashing(M) \equiv$ {

[1] Автор шифра: *Ronald Rivest* (США, компания RSA Data Security).

$j := M_2 \& 63;\ M_3 := M_3 - k_j;$
$j := M_1 \& 63;\ M_2 := M_2 - k_j;$
$j := M_0 \& 63;\ M_1 := M_1 - k_j;$
$j \coloneqq M3 \& 63;\ M_0 \coloneqq M_0 - k_j$
}.

Алгоритм зашифрования *RC2*

Вход: M − 64-битовый блок открытых данных.
for $j := 0,4,8,12,16$ **do** $Emixing(M, j)$;
$Emashing(M)$;
for $j := 20,24,28,32,36,40$ **do** $Emixing(M, j)$;
$Emashing(M)$;
for $j := 44,48,52,56,60$ **do** $Emixing(M, j)$.
Выход: шифртекст M.

Для расшифрования шифртекста необходимо выполнить обратные преобразования в обратном порядке.

$RC5$. Криптоалгоритм $RC5$ [2] представляет собой семейство алгоритмов блочного шифрования, определяемое следующими параметрами:
w − размер слова в битах ($RC5$ шифрует данные блоками длиной в 2 слова);
r − число раундов шифрования;
b − число байтов в секретном ключе K.
Указанные параметры могут варьироваться в следующих пределах: $w = 16$, 32 или 64; $0 \leq r, b \leq 255$. Конкретная версия алгоритма обозначается $RC5$-$w/r/b$. Например, $RC5$-32/12/16 использует 32-битовые слова (или 64-битовые блоки открытых данных и шифртекста), 12 раундов шифрования и 16-байтовый (128-битовый) секретный ключ. Данная версия считается стандартной. При упаковке байтов в слова соблюдается соглашение о $little\text{-}endian$ - порядке байтов: первый байт является младшим.

В алгоритме используются операции над w-битовыми словами A и B:

$A + B$ и $A - B$ − сложение и вычитание по модулю 2^w;
$A \oplus B$ − побитовое сложение по модулю 2;
$rol(A, B)$ и $ror(A, B)$ − циклические сдвиги влево и вправо на n битовых позиций, где $n = B \bmod w$.

На этапе предвычислений секретный ключ K преобразуется в расширенный ключ $Q = (Q_0, Q_1, \ldots, Q_{2r+1})$, состоящий из $2r + 2$ w-битовых подключей Q_i.

Алгоритм зашифрования *RC5*

Вход: $P = L \parallel R$ − $2w$-битовый блок открытых данных, представленный в виде конкатенации -битовых слов L и R.
$L := L + Q_0$;
$R := R + Q_1$;
for $i := 1$ **to** r **do** {
$L := rol(L \oplus R, R) + Q_{2i}$;
$R := rol(L \oplus R, L) + Q_{2i+1}$
}.
Выход: $C = L \parallel R$ − $2w$-битовый блок шифртекста.

Алгоритм расшифрования *RC5*

Вход: $C = L \parallel R$ − $2w$-битовый блок шифртекста.
for $i := r$ **downto** 1 **do** {

[2] Автор шифра: *Ronald Rivest* (США, компания RSA Data Security).

$R := ror(R - Q_{2i+1}, L) \oplus L$;
$L := ror(L - Q_{2i}, R) \oplus R$
};
$R := R - Q_1$;
$L := L - Q_0$.
Выход: $P = L \parallel R$ – $2w$-битовый блок открытых данных.

Расширенный ключ Q формируется на основе исходного -байтового секретного ключа $K = (k_0, k_1, \ldots, k_{b-1})$:

Массив Q инициализируется псевдослучайными значениями. Для этого используются две w-битовые константы:

$C_w = Odd[(e-2)2^w]$,
$G_w = Odd[(\phi-2)2^w]$,

где e – основание натуральных логарифмов, $\phi = (1+\sqrt{5})/2$ – отношение золотого сечения, а $Odd(x)$ – нечетное целое, ближайшее к x. Для допустимых значений w эти константы в 16-ичном представлении будут следующими:

$C_{16} = 0\text{x}b7e1$,
$C_{32} = 0\text{x}b7e15163$,
$C_{64} = 0\text{x}b7e151628aed2a6b$;
$G_{16} = 0\text{x}9e37$,
$G_{32} = 0\text{x}9e3779b9$,
$G_{64} = 0\text{x}9e3779b97f4a7c15$.

С использованием этих констант инициализация массива Q проводится по схеме:
$Q_0 := C_w$;
for $i := 1$ **to** $2r+1$ **do** $Q_i := Q_{i-1} + G_w$.

Секретный ключ K преобразуется в массив w-битовых слов $L = (L_0, L_1, \ldots, L_{c-1})$, где $c = (b + w - 1)\ div\ w$. Для этого массив L сначала обнуляется, а затем ключ K копируется в L. (Если $8b$ не кратно w, то в L правые байты остаются нулевыми.) Наконец, массив Q смешивается с массивом L:

$i := j := 0$;
$X := Y := 0$;
$t := 2r + 2$;
for $k := 1$ **to** $3 \cdot max(t, c)$ **do** {
$Q_i := rol(Q_i + X + Y, 3)$;
$X := Q_i$;
$i := (i+1)\ mod\ t$;
$L_j := rol(L_j + X + Y, X + Y)$;
$Y := L_j$;
$j := (j+1)\ mod\ c$
}.
Расширенный ключ Q создан.

RC6. Криптоалгоритм $RC6$ [3], шифрует 128-битовые блоки открытых данных под управлением секретного ключа, длина которого может варьироваться от 4 до 32 байтов (с шагом 4).

В алгоритме используются элементарные операции над 32-битовыми словами: сложение $(+)$, вычитание $(-)$ и умножение $(\times)$ по модулю 2^{32}; побитовое сложение по модулю 2 $(\oplus)$; циклические сдвиги на s битов влево (rol_s) и вправо (ror_s).

[3] Авторы шифра: *Ronald L. Rivest*, *Matt Robshaw*, *Ray Sidney* и *Yigun Lisa Yin* (США).

Исходный секретный ключ K, представленный в виде с 32-битовых слов q_0, q_1,…, q_{c-1} разворачивается в 44 раундовых подключа k_0, k_1,…, k_{43}:

$k_0 := 0\text{x}b7e15163$;
$\boldsymbol{for}\ i := 1\ \boldsymbol{to}\ 43\ \boldsymbol{do}\ k_i := k_{i-1} + 0\text{x}9e3779b9$;
$a := b := 0$;
$i := j := 0$;
$\boldsymbol{for}\ s := 1\ \boldsymbol{to}\ 132\ \boldsymbol{do}$ {
    $a := rol_3(k_i + a + b)$;
    $k_i := a$;
    $b := rol_{a+b}(q_j + a + b)$;
    $q_j := b$;
    $i := (i + 1)\ mod\ 44$;
    $j := (j + 1)\ mod\ c$
}.

Алгоритм зашифрования $RC6$

Вход: $P = (a, b, c, d) -$ 128-битовый блок открытых данных в виде четырех 32-битовых слов a, b, c, d.

$b := b + k_0$;
$d := d + k_1$;
$\boldsymbol{for}\ i := 1\ \boldsymbol{to}\ 20\ \boldsymbol{do}$ {
    $t := rol_5(b \times (2 \times b + 1))$;
    $u := rol_5(d \times (2 \times d + 1))$;
    $a := rol_u(a \oplus t) + k_{2i}$;
    $c := rol_t(c \oplus u) + k_{2i+1}$;
    $(a, b, c, d) := (b, c, d, a)$
};
$a := a + k_{42}$;
$c := c + k_{43}$.

Выход: $P = (a, b, c, d) -$ 128-битовый блок шифртекста.