## *SC* 2000

Криптоалгоритм $SC$ 2000 [1] шифрует 128-битовые блоки открытых данных под управлением секретного ключа, длина которого может составлять 128, 192 или 256 битов.

### Обозначения

$a, b, c, d, e, f, g, h, mask$ – 32-битовые слова (блоки);

$a_i$ – $i$-ый бит в слове $a$; $a = a_0 \| a_1 \| \ldots \| a_{31}$, где $a_0$ – старший, $a_{31}$ – младший биты;

$a_{k..m}$, где $k \le m$, – блок, составленный из битов $a_k, a_{k+1}, \ldots, a_m$, т.е. $a_{k..m} = a_k \| a_{k+1} \| \cdots \| a_m$;

$KeyLength$ – длина $(128/192/256)$ секретного ключа $K$.

$ek[0..63]$ – массив из 64 32-битовых раундовых подключей, генерируемых на основе секретного ключа $K$ (если $KeyLength = 128$, то используются только подключи $ek[0..55]$).

### Операции над 32- битовыми словами

$\oplus$ $(xor)$ – побитовое сложение по модулю 2;

$\&$ $(and)$ – побитовое умножение;

$\vee$ $(or)$ – побитовая дизъюнкция;

$\boxplus_{32}$ – сложение по модулю $2^{32}$;

$\boxminus_{32}$ – вычитание по модулю $2^{32}$;

$\boxtimes_{32}$ – умножение по модулю $2^{32}$;

$rol_1(a)$ – циклический сдвиг слова $a$ влево на 1 бит.

### Функции, используемые в *SC* 2000

Функция $Rf\ (a, b, c, d, mask)$ (см. рис.1) возвращает значение $(e, f, g, h)$:

$(s, t) := Ff(c, d, mask)$;
$(g, h) := (a \oplus s, b \oplus t)$;
$(s, t) := Ff(g, h, mask)$;
$(e, f) := (c \oplus s, d \oplus t)$.

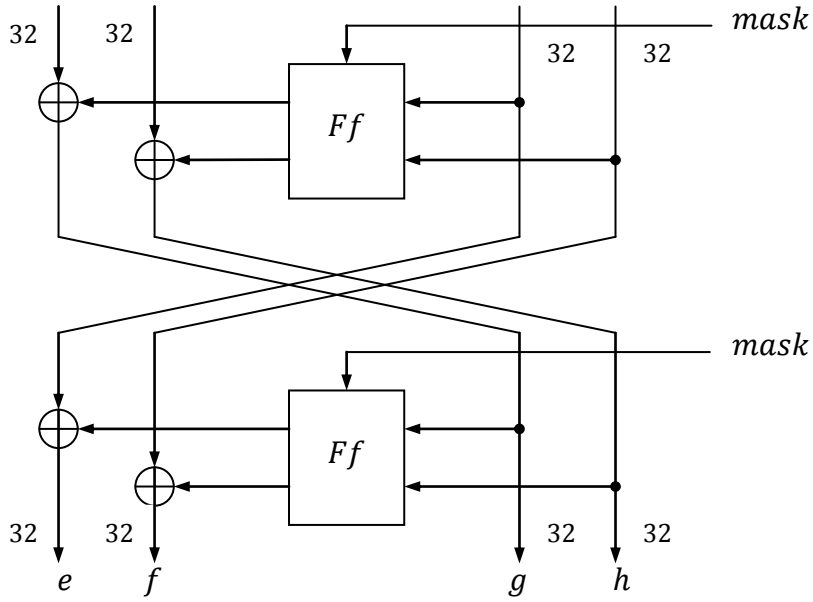


Рис. 1. *Функция* $Rf$

Функция $Ff\ (a, b, mask)$ (см. рис. 2) возвращает значение

$(c, d) = Lf(M(S(a)), M(S(b)), mask)$.

---

[1] Авторы шифра: *Shimoyama, H. Yanami, K. Yokoyama, M. Takenaka, K. Itoh, N. Torii, H. Yajima* (Япония)

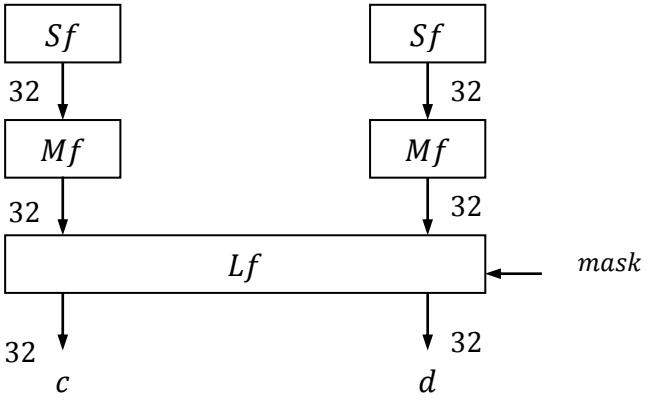


Рис. 2. *Функция* $Ff$

Функция $Sf\ (a)$ (см. рис. 3) возвращает 32-битовое значение

$S_6(a_{0..5})||S_5(a_{6..10})||S_5(a_{11..15})||S_5(a_{16..20})||S_5(a_{21..25})||S_6(a_{26..31})$,

где $S_6$ и $S_5$ – подстановки соответственно на множествах 6-битовых и 5-битовых подблоков, заданные таблицей 1.

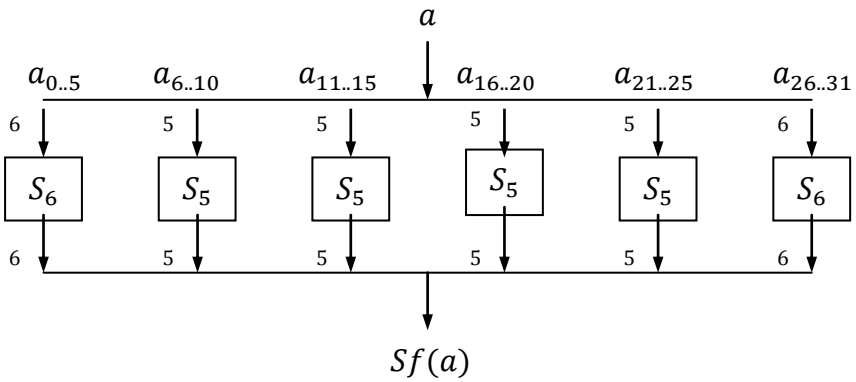


Рис. 3. *Функция* $Sf$

Функция $Mf(a)$ возвращает 32-битовое значение $b$:

$b := 0;$

$\boldsymbol{for}\ i := 0\ \boldsymbol{to}\ 31\ \boldsymbol{do\ if}\ a_i = 1\ \boldsymbol{then}\ b := b \oplus M_i,$

где $M[0..31]$ – массив 32-битовых констант, заданный таблицей 2.

Функция $Lf\ (a, b, mask)$ (см. рис. 4) возвращает значение

$(c, d) = (b \oplus (a\ \&\ mask), a \oplus (b\ \&\ (not\ mask)))$.

Функция $Bf\ (a, b, mask)$ возвращает значение $(e, f, g, h)$, вычисляемое следующим образом:

$\boldsymbol{for}\ i := 0\ \boldsymbol{to}\ 31\ \boldsymbol{do}\ e_i \parallel f_i \parallel g_i \parallel h_i := S_4(a_i \parallel b_i \parallel c_i \parallel d_i),$

где $S_4$ – подстановка на множестве 4-битовых подблоков полубайтов, заданная табл. 1.

(*Пояснение*: полубайт $x^{(4)}$, образованный очередной четверкой битов $a_i$, $b$, $c_i$ и $d_i$, преобразуется с помощью подстановки $S_4$ в полубайт $y^{(4)} = S_4(x^{(4)})$, биты которого задают соответствующие биты результата – слов $e, f, g, h$. Обратная функция $Bf^{-1}$ задается аналогично – с заменой подстановки $S_4$ на $S_4^{-1}$. Функция $Gf\ (a, b, c, d)$ и $Wf\ (a, b, c, d)$ возвращают 32-битовые значения:

$Gf(a, b, c, d) \equiv (rol_1(a) \boxplus_{32} b) \oplus rol_1(rol_1(c) \boxminus_{32} d)$.

$Wf(a, b, c, d) \equiv Mf(Sf((Mf(Sf(a)) \boxplus_{32} Mf(Sf(b))) \oplus (Mf(Sf(c)) \boxtimes_{32} d)))$.

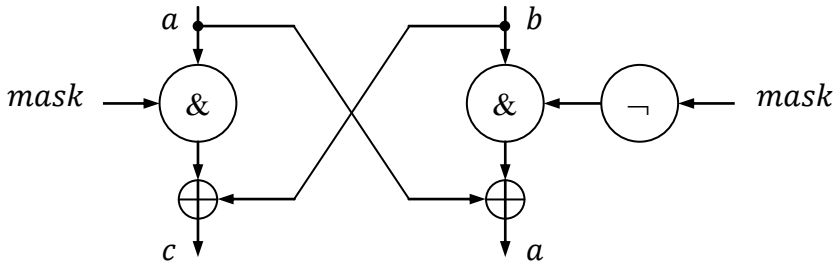


Рис. 4. *Функция Lf*

### Генерация раундовых подключей в *SC* 2000

Секретный ключ $K$ записывается в массив 32-битовых слов $uk[0]$, $uk[1]$,…, uk[7]:

$\boldsymbol{if}\ KeyLength = 128\ \boldsymbol{then}\ \{$
$\quad uk[0..3] := K;$
$\quad uk[4..7] := uk[0..3]$
$\}$
$\boldsymbol{else\ if}\ KeyLength = 192\ \boldsymbol{then}\ \{$
$\quad uk[0..5] := K;$
$\quad uk[6..7] := uk[0..1]$
$\quad \}$
$\quad \boldsymbol{else}\ uk[0..7] := K.$

При вычислении раундовых подключей $ek[]$ используется массив $Index\ [0..8][0..3]$, заданный табл. 3 и массивы вспомогательных 32-битовых переменных $aa[0..2]$, $bb[0..2]$, $cc[0..2]$ и $dd[0..2]$.

$\boldsymbol{for}\ i := 0\ \boldsymbol{to}\ 2\ \boldsymbol{do}\ \{$
$\quad aa[i] := Wf(4i, uk[0], uk[1], i+1);$
$\quad bb[i] := Wf(4i+1, uk[2], uk[3], i+1);$
$\quad cc[i] := Wf(4i+2, uk[4], uk[5], i+1);$
$\quad dd[i] := Wf(4i+3, uk[6], uk[7], i+1)$
$\};$
$\boldsymbol{if}\ KeyLength = 128\ \boldsymbol{then}\ num\_ekey := 55\ \boldsymbol{else}\ num\_ekey := 63;$
$\boldsymbol{for}\ n := 0\ \boldsymbol{to}\ num\_ekey\ \boldsymbol{do}\ \{$
$\quad u := n\ \boldsymbol{mod}\ 9;$
$\quad v := (n + (n\ \boldsymbol{div}\ 36))\ mod\ 12;$
$\quad x := Index[u][0];$
$\quad y := Index[u][1];$
$\quad z := Index[u][2];$
$\quad w := Index[u][3];$
$\quad \boldsymbol{case}\ v\ \boldsymbol{of}$
$\quad\quad 0: \{(a,b,c,d) := (aa[x], bb[y], cc[z], dd[w])\};$
$\quad\quad 1: \{(a,b,c,d) := (bb[x], aa[y], dd[z], cc[w])\};$
$\quad\quad 2: \{(a,b,c,d) := (cc[x], dd[y], aa[z], bb[w])\};$
$\quad\quad 3: \{(a,b,c,d) := (dd[x], cc[y], bb[z], aa[w])\};$
$\quad\quad 4: \{(a,b,c,d) := (aa[x], cc[y], dd[z], bb[w])\};$
$\quad\quad 5: \{(a,b,c,d) := (bb[x], dd[y], cc[z], aa[w])\};$
$\quad\quad 6: \{(a,b,c,d) := (cc[x], aa[y], bb[z], dd[w])\};$
$\quad\quad 7: \{(a,b,c,d) := (dd[x], bb[y], aa[z], cc[w])\};$
$\quad\quad 8: \{(a,b,c,d) := (aa[x], dd[y], bb[z], cc[w])\};$
$\quad\quad 9: \{(a,b,c,d) := (bb[x], cc[y], aa[z], dd[w])\};$
$\quad\quad 10: \{(a,b,c,d) := (cc[x], bb[y], dd[z], aa[w])\};$

11: $\{(a,b,c,d) := (dd[x], aa[y], cc[z], bb[w])\}$

***end case***;

$ek[n] := Gf(a,b,c,d)$

}.

Таблица 1

*Подстановки* $S_6$, $S_5$, $S_4$ *и* $S_4^{-1}$

| $S_6[0..63]$ | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 47 | 59 | 25 | 42 | 16 | 23 | 28 | 39 | 26 | 38 | 36 | 19 | 60 | 24 | 39 | 56 |
| 37 | 63 | 20 | 61 | 56 | 02 | 30 | 44 | 08 | 10 | 06 | 22 | 53 | 47 | 51 | 11 |
| 62 | 52 | 35 | 18 | 14 | 46 | 00 | 54 | 17 | 40 | 27 | 04 | 31 | 08 | 05 | 12 |
| 03 | 16 | 41 | 34 | 33 | 07 | 45 | 49 | 50 | 58 | 01 | 21 | 43 | 57 | 32 | 13 |
| $S_5[0..31]$ | | | | | | | | | | | | | | | |
| 20 | 26 | 07 | 31 | 19 | 12 | 10 | 15 | 22 | 30 | 13 | 14 | 04 | 24 | 09 | 18 |
| 27 | 11 | 01 | 21 | 06 | 16 | 02 | 28 | 23 | 05 | 08 | 03 | 00 | 17 | 29 | 25 |
| $S_4[0..15]$ | | | | | | | | | | | | | | | |
| 02 | 05 | 10 | 12 | 07 | 15 | 01 | 11 | 13 | 06 | 00 | 09 | 04 | 08 | 03 | 14 |
| $S_4^{-1}[0..15]$ | | | | | | | | | | | | | | | |
| 10 | 06 | 00 | 14 | 12 | 01 | 09 | 04 | 13 | 11 | 02 | 07 | 03 | 08 | 15 | 05 |

Таблица 2

*Массив* $M[0..31]$

| | | | |
|---|---|---|---|
| d0c19225 | a5a2240a | 1b84d250 | b728a4a1 |
| 6a704902 | 85dddbe6 | 766ff4a4 | ecdfe128 |
| afd13e94 | df837d09 | bb27fa52 | 695059ad |
| 52a1bb58 | cc322f1d | 1844565b | b4a8acf6 |
| 34235438 | 6847a851 | e48c0cbb | cd181136 |
| 9a112a0c | 43ec6d0e | 87d8d27d | 487dc995 |
| 90fb9b4b | a1f63697 | fc513ed9 | 78a37d93 |
| 8d16c5df | 9e0c8bbe | 3c381f7c | e9fb0779 |

Таблица 3

*Массив Index*

| $u$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| $Index[u][0]$ | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 |
| $Index[u][1]$ | 0 | 1 | 2 | 1 | 2 | 0 | 2 | 0 | 1 |
| $Index[u][2]$ | 0 | 1 | 2 | 0 | 1 | 2 | 0 | 1 | 2 |
| $Index[u][3]$ | 0 | 1 | 2 | 1 | 2 | 0 | 2 | 0 | 1 |

## Алгоритм зашифрования *SC* 2000

(см. рис. 5)

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

$c0 := 0x55555555;$

$c1 := 0x3333333;$

$(e,f,g,h) := (a,b,c,d);$

***for*** $i := 0$ ***to*** $5$ ***do*** {

$(e,f,g,h) := (e \oplus ek[8i], f \oplus ek[8i+1], g \oplus ek[8i+2], h \oplus ek[8i+3]);$

$(e,f,g,h) := Bf(e,f,g,h);$

$(e,f,g,h) := (e \oplus ek[8i+4], f \oplus ek[8i+5], g \oplus ek[8i+6], h \oplus ek[8i+7]);$

$(e,f,g,h) := Rf(e,f,g,h,c0);$

$c0 \leftrightarrow c1$

};

$(e, f, g, h) := (e \oplus ek[48], f \oplus ek[49], g \oplus ek[50], h \oplus ek[51])$;

$(e, f, g, h) := Bf(e, f, g, h)$;

$(e, f, g, h) := (e \oplus ek[52], f \oplus ek[53], g \oplus ek[54], h \oplus ek[55])$;

$\boldsymbol{if}\ KeyLength \neq 128\ \boldsymbol{then}$ {

$(e, f, g, h) := Rf(e, f, g, h, c0)$;

$(e, f, g, h) := (e \oplus ek[56], f \oplus ek[57], g \oplus ek[58], h \oplus ek[59])$;

$(e, f, g, h) := Bf(e, f, g, h)$;

$(e, f, g, h) := (e \oplus ek[60], f \oplus ek[61], g \oplus ek[62], h \oplus ek[63]$

}.

*Выход*: $C = (e, f, g, h) -$ 128-битовый блок шифртекста в виде четырех 32-битовых слов $e, f, g, h$.

## Алгоритм расшифрования *SC* 2000

(см. рис. 5)

*Вход*: $C = (e, f, g, h) -$ 128-битовый блок шифртекста в виде четырех 32-битовых слов $e, f, g, h$.

$c0 := 0x55555555$;

$c1 := 0x33333333$;

$(a, b, c, d) := (e, f, g, h)$;

$\boldsymbol{if}\ KeyLength \neq 128\ \boldsymbol{then}$ {

$(a, b, c, d) := (a \oplus ek[60], b \oplus ek[61], c \oplus ek[62], d \oplus ek[63])$;

$(a, b, c, d) := Bf^{-1}(a, b, c, d)$;

$(a, b, c, d) := (a \oplus ek[56], b \oplus ek[57], c \oplus ek[58], d \oplus ek[59])$;

$(a, b, c, d) := Rf(a, b, c, d, c0)$;

$\boldsymbol{for}\ i := 6\ \boldsymbol{downto}\ 1\ \boldsymbol{do}$ {

$(a, b, c, d) := (a \oplus ek[8i + 4], b \oplus ek[8i + 5], c \oplus ek[8i + 6], d \oplus ek[8i + 7])$;

$(a, b, c, d) := Bf^{-1}(a, b, c, d)$;

$(a, b, c, d) := (a \oplus ek[8i], b \oplus ek[8i + 1], c \oplus ek[8i + 2], d \oplus ek[8i + 3])$;

$c0 \leftrightarrow c1$

};

$(a, b, c, d) := Rf(a, b, c, d, c0)$;

$(a, b, c, d) := (a \oplus ek[4], b \oplus ek[5], c \oplus ek[6], d \oplus ek[7])$;

$(a, b, c, d) := Bf^{-1}(a, b, c, d)$;

$(a, b, c, d) := (a \oplus ek[0], b \oplus ek[1], c \oplus ek[2], d \oplus ek[3])$;

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

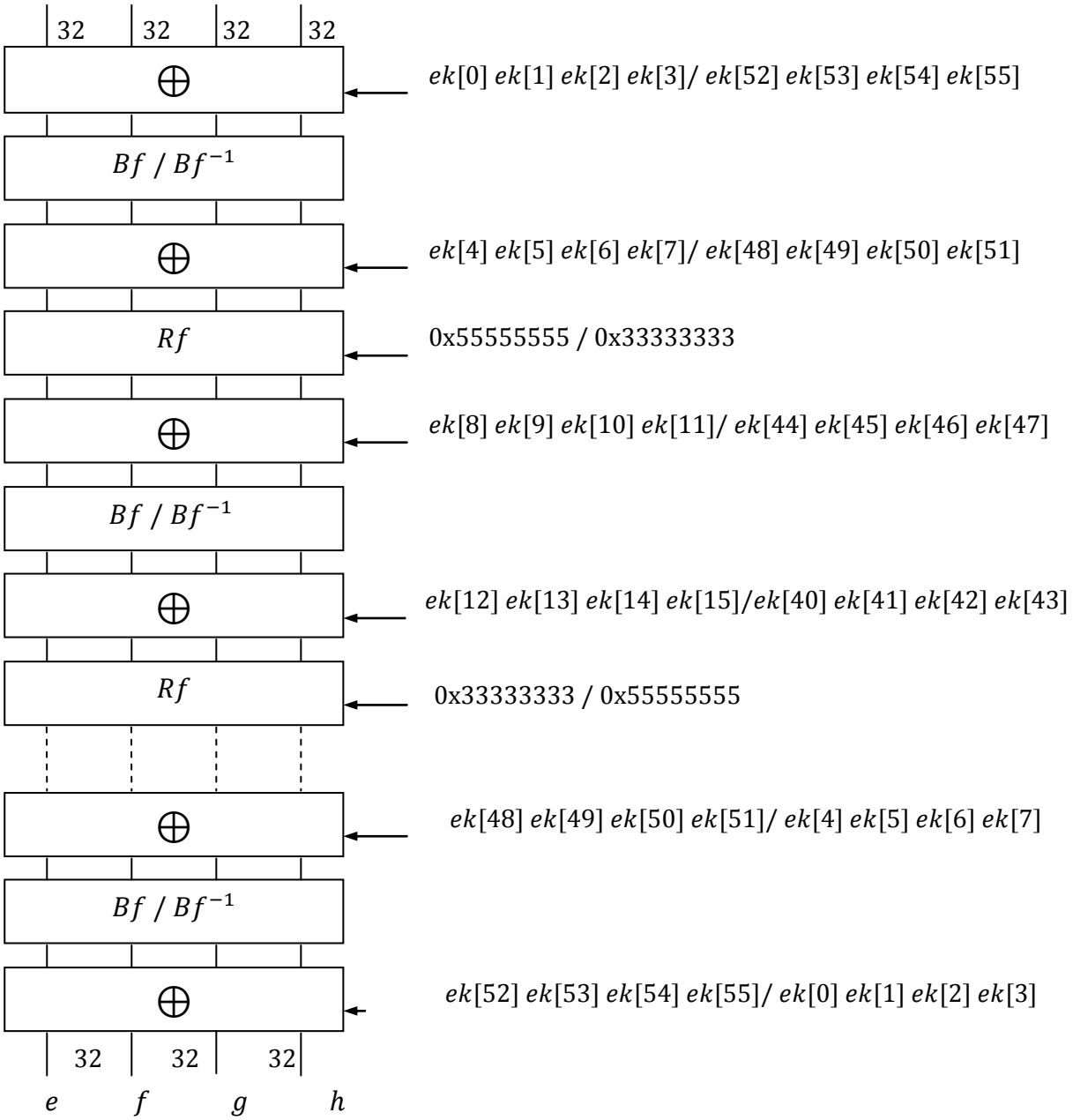


Рис. 5. *Функция зашифрования/расшифрования SC* 2000