Базовый блочный шифр

В настоящем документе приведено описание алгоритма блочного шифрования с длиной блока 128 бит и длиной ключа 256 бит.

1 Обозначения

В настоящем описании используются следующие обозначения:

V^*	множество всех двоичных строк конечной размерности, включая пустую строку;
V_n	множество всех двоичных строк размерности n, где n – целое неотрицательное число. Нумерация подстрок и компонент строки осуществляется справа налево начиная с нуля;
$\|A\|$	размерность (число компонент) строки $A \in V^*$;
$A\|\|B$	конкатенация строк $A, B \in V^*$, т.е. строка из $V_{\|A\|+\|B\|}$, в которой левая подстрока из $V_{\|A\|}$ совпадает со строкой A, а правая подстрока из $V_{\|B\|}$ совпадает со строкой B;
$\oplus$	операция покомпонентного сложения по модулю 2 двух двоичных строк одинаковой размерности;
Z_m	множество $\{0, 1, \ldots, m-1\}$;
P	конечное поле $GF(2)[x]/g(x)$, где $g(x) = x^8 + x^7 + x^6 + x + 1 \in GF(2)[x]$. Элементы поля P представляются целыми числами, причем числу $b_0 + b_1 \cdot 2 + ... + b_7 \cdot 2^7 \in Z_{2^8}$, $b_i \in \{0, 1\}$, $i = 0, 1, \ldots, 7$, соответствует элемент $b_0 + b_1 \cdot \theta + ... + b_7 \cdot \theta^7 \in P$, где θ обозначает класс вычетов по модулю $g(x)$, содержащий x.

$\mathcal{V}_n : Z_{2^n} \to V_n$	биективное отображение, сопоставляющее каждому элементу z множества Z_{2^n}, где $z = z_0 + 2 \cdot z_1 + \ldots + 2^{n-1} \cdot z_{n-1}$, $z_j \in \{0, 1\}$, $j = 0, 1, \ldots, n-1$, его двоичное представление, т. е. выполняется равенство $\mathcal{V}_n(z) = z_{n-1} \| \ldots \| z_1 \| z_0$;
$\mathcal{I}_n : V_n \to Z_{2^n}$	отображение, обратное к отображению $\mathcal{V}_n$, т. е. $\mathcal{I}_n = \mathcal{V}_n^{-1}$;
$\Delta : V_8 \to P$	биективное отображение, сопоставляющее двоичной строке из V_8 элемент поля P следующим образом: строке $b_7 \| b_6 \| \ldots \| b_0$, $b_i \in V_1$, $i = 0, 1, \ldots, 7$, соответствует элемент $b_0 + b_1 \cdot \theta + \ldots + b_7 \cdot \theta^7 \in P$;
$\nabla : P \to V_8$	отображение, обратное к отображению Δ, т.е. $\nabla = \Delta^{-1}$;
$\Phi\Psi$	композиция отображений, при которой отображение Ψ действует первым.

2 Описание алгоритма блочного шифрования

2.1 Значения параметров

2.1.1 Нелинейное биективное преобразование

В качестве нелинейного биективного преобразования выступает подстановка $\pi = \mathcal{V}_8 \pi' \mathcal{I}_8 : V_8 \to V_8$, где $\pi' : Z_{2^8} \to Z_{2^8}$. Значения подстановки π' записаны ниже в виде массива $\pi' = (\pi'(0), \pi'(1), \ldots, \pi'(255))$:

$\pi' =$ (252, 238, 221, 17, 207, 110, 49, 22, 251, 196, 250, 218, 35, 197, 4, 77, 233, 119, 240, 219, 147, 46, 153, 186, 23, 54, 241, 187, 20, 205, 95, 193, 249, 24, 101, 90, 226, 92, 239, 33, 129, 28, 60, 66, 139, 1, 142, 79, 5, 132, 2, 174, 227, 106, 143, 160, 6, 11, 237, 152, 127, 212, 211, 31, 235, 52, 44, 81, 234, 200, 72, 171, 242, 42, 104, 162, 253, 58, 206, 204, 181, 112, 14, 86, 8, 12, 118, 18, 191, 114, 19, 71, 156,

183, 93, 135, 21, 161, 150, 41, 16, 123, 154, 199, 243, 145, 120, 111, 157, 158, 178, 177, 50, 117, 25, 61, 255, 53, 138, 126, 109, 84, 198, 128, 195, 189, 13, 87, 223, 245, 36, 169, 62, 168, 67, 201, 215, 121, 214, 246, 124, 34, 185, 3, 224, 15, 236, 222, 122, 148, 176, 188, 220, 232, 40, 80, 78, 51, 10, 74, 167, 151, 96, 115, 30, 0, 98, 68, 26, 184, 56, 130, 100, 159, 38, 65, 173, 69, 70, 146, 39, 94, 85, 47, 140, 163, 165, 125, 105, 213, 149, 59, 7, 88, 179, 64, 134, 172, 29, 247, 48, 55, 107, 228, 136, 217, 231, 137, 225, 27, 131, 73, 76, 63, 248, 254, 141, 83, 170, 144, 202, 216, 133, 97, 32, 113, 103, 164, 45, 43, 9, 91, 203, 155, 37, 208, 190, 229, 108, 82, 89, 166, 116, 210, 230, 244, 180, 192, 209, 102, 175, 194, 57, 75, 99, 182).

2.1.2 Линейное преобразование

Линейное преобразование задается отображением $l : V_8^{16} \to V_8$, которое определяется следующим образом:

$l(a_{15}, \ldots, a_0) = \nabla(148 \cdot \Delta(a_{15}) + 32 \cdot \Delta(a_{14}) + 133 \cdot \Delta(a_{13}) + 16 \cdot \Delta(a_{12}) + 194 \cdot \Delta(a_{11}) + 192 \cdot \Delta(a_{10}) + 1 \cdot \Delta(a_9) + 251 \cdot \Delta(a_8) + 1 \cdot \Delta(a_7) + 192 \cdot \Delta(a_6) + 194 \cdot \Delta(a_5) + 16 \cdot \Delta(a_4) + 133 \cdot \Delta(a_3) + 32 \cdot \Delta(a_2) + 148 \cdot \Delta(a_1) + 1 \cdot \Delta(a_0))$

для любых $a_i \in V_8$, $i = 0, 1, \ldots, 15$, где операции сложения и умножения осуществляются в поле P.

2.2 Преобразования

При реализации алгоритма шифрования используются следующие преобразования:

$X[k] : V_{128} \to V_{128}$, $X[k](a) = k \oplus a$, $k, a \in V_{128}$;

$S : V_{128} \to V_{128}$, $S(a) = S(a_{15} \| \ldots \| a_0) = \pi(a_{15}) \| \ldots \| \pi(a_0)$,
где $a = a_{15} \| \ldots \| a_0 \in V_{128}$, $a_i \in V_8$, $i = 0, 1, \ldots, 15$;

$S^{-1} : V_{128} \to V_{128}$, преобразование, обратное к преобразованию S, которое может быть вычислено, например, следующим образом:
$S^{-1}(a) = S^{-1}(a_{15}\| \dots \|a_0) = \pi^{-1}(a_{15})\| \dots \|\pi^{-1}(a_0)$,
где $a = a_{15}\| \dots \|a_0 \in V_{128}$, $a_i \in V_8$, $i = 0, 1, \dots, 15$,
π^{-1} – подстановка, обратная к подстановке π;

$R : V_{128} \to V_{128}$, $R(a) = R(a_{15}\| \dots \|a_0) = l(a_{15}, \dots, a_0)\|a_{15}\| \dots \|a_1$,
где $a = a_{15}\| \dots \|a_0 \in V_{128}$, $a_i \in V_8$, $i = 0, 1, \dots, 15$;

$L : V_{128} \to V_{128}$, $L(a) = R^{16}(a)$;

$R^{-1} : V_{128} \to V_{128}$, преобразование, обратное к преобразованию R, которое может быть вычислено, например, следующим образом: $R^{-1}(a) = R^{-1}(a_{15}\| \dots \|a_0) =$
$= a_{14}||a_{13}|| \dots ||a_0||l(a_{14}, a_{13}, \dots, a_0, a_{15})$,
где $a = a_{15}\| \dots \|a_0 \in V_{128}$, $a_i \in V_8$, $i = 0, 1, \dots, 15$;

$L^{-1} : V_{128} \to V_{128}$, $L^{-1}(a) = (R^{-1})^{16}(a)$;

$F[C] : V_{128}^2 \to V_{128}^2$, $F[C](a_1, a_0) = (LSX[C](a_1) \oplus a_0, a_1)$, где $C \in V_{128}$, $a_0, a_1 \in V_{128}$.

2.3 Алгоритм развертки ключа

Алгоритм развертки ключа использует итерационные константы $C_i \in V_{128}$, $i = 1, 2, \dots, 32$, которые определены следующим образом:

$$C_i = L(\mathcal{V}_{128}(i)), \quad i = 1, 2, \dots, 32.$$

Итерационные ключи $K_i \in V_{128}$, $i = 1, 2, \dots, 10$, вырабатываются на основе мастер-ключа $K \in V_{256}$ и определяются равенствами:

$$K_1\|K_2 = K;$$

$$(K_{2i+1}, K_{2i+2}) = F[C_{8(i-1)+8}] \dots F[C_{8(i-1)+1}](K_{2i-1}, K_{2i}), \quad i = 1, 2, 3, 4.$$

2.4 Базовый алгоритм шифрования

Базовый алгоритм шифрования реализует перестановку элементов множества V_{128} в зависимости от значений итерационных ключей $K_i \in V_{128}$, $i = 1, 2, \ldots, 10$.

2.4.1 Алгоритм зашифрования

Алгоритм зашифрования реализует преобразование множества V_{128} в соответствии с равенством

$$E_{K_1,\ldots,K_{10}}(a) = X[K_{10}]LSX[K_9]\ldots LSX[K_2]LSX[K_1](a),$$

где $a \in V_{128}$.

2.4.2 Алгоритм расшифрования

Алгоритм расшифрования реализует преобразование множества V_{128} в соответствии с равенством

$$D_{K_1,\ldots,K_{10}}(a) = X[K_1]S^{-1}L^{-1}X[K_2]\ldots S^{-1}L^{-1}X[K_9]S^{-1}L^{-1}X[K_{10}](a),$$

где $a \in V_{128}$.

3 Контрольные примеры

Для удобства, в дальнейшем изложении двоичные строки из V^*, длина которых кратна 4, будем записывать в шестнадцатеричном виде, а символ конкатенации (“||”) будем опускать. То есть, строка $A \in V_{4n}$, будет представлена в виде

$$a_{n-1}a_{n-2}\ldots a_0,$$

где $a_i \in \{0, 1, \ldots, 9, a, b, c, d, e, f\}$. Соответствие между двоичными строками длины 4 и шестнадцатеричными строками длины 1 задаётся естественным образом, см. таблицу 3. Преобразование, ставящее в соответствие двоичной строке длины $4n$ шестнадцатеричную строку длины n, и соответствующее обратное преобразование, для простоты записи будем опускать.

Таблица 3: Соответствие между двоичными и шестнадцатеричными строками

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	a
1011	b
1100	c
1101	d
1110	e
1111	f

3.1 Контрольные примеры для преобразования S

$$S(\text{ffeeddccbbaa99881122334455667700}) = \text{b66cd8887d38e8d77765aeea0c9a7efc},$$

$$S(\text{b66cd8887d38e8d77765aeea0c9a7efc}) = \text{559d8dd7bd06cbfe7e7b262523280d39},$$

$$S(\text{559d8dd7bd06cbfe7e7b262523280d39}) = \text{0c3322fed531e4630d80ef5c5a81c50b},$$

$$S(\text{0c3322fed531e4630d80ef5c5a81c50b}) = \text{23ae65633f842d29c5df529c13f5acda}.$$

3.2 Контрольные примеры для преобразования R

$$R(\text{00000000000000000000000000000100}) = \text{94000000000000000000000000000001},$$

$$R(\text{94000000000000000000000000000001}) = \text{a5940000000000000000000000000000},$$

$$R(\text{a5940000000000000000000000000000}) = \text{64a5940000000000000000000000000000},$$

$$R(\text{64a5940000000000000000000000000000}) = \text{0d64a5940000000000000000000000000000}.$$

3.3 Контрольные примеры для преобразования L

$$L(\text{64a594000000000000000000000000000}) = \text{d456584dd0e3e84cc3166e4b7fa2890d},$$

$$L(\text{d456584dd0e3e84cc3166e4b7fa2890d}) = \text{79d26221b87b584cd42fbc4ffea5de9a},$$

$$L(\text{79d26221b87b584cd42fbc4ffea5de9a}) = \text{0e93691a0cfc60408b7b68f66b513c13},$$

$$L(\text{0e93691a0cfc60408b7b68f66b513c13}) = \text{e6a8094fee0aa204fd97bcb0b44b8580}.$$

3.4 Контрольный пример для алгоритма развертки ключа

В настоящем контрольном примере мастер-ключ имеет значение

K=8899aabbccddeeff0011223344556677fedcba98765432100123456789abcdef,

$$(K_1, K_2) =$$

=(8899aabbccddeeff0011223344556677,fedcba98765432100123456789abcdef),

$C_1 = \text{6ea276726c487ab85d27bd10dd849401},$
$X[C_1](K_1) = \text{e63bdcc9a09594475d369f2399d1f276},$
$SX[C_1](K_1) = \text{0998ca37a7947aabb78f4a5ae81b748a},$
$LSX[C_1](K_1) = \text{3d0940999db75d6a9257071d5e6144a6},$

$F[C_1](K_1, K_2) =$
$= (\text{c3d5fa01ebe36f7a9374427ad7ca8949,8899aabbccddeeff0011223344556677}),$

$C_2 = \text{dc87ece4d890f4b3ba4eb92079cbeb02},$
$F[C_2]F[C_1](K_1, K_2) =$
$= (\text{37777748e56453377d5e262d90903f87,c3d5fa01ebe36f7a9374427ad7ca8949}),$

$C_3 = \text{b2259a96b4d88e0be7690430a44f7f03},$
$F[C_3] \dots F[C_1](K_1, K_2) =$
$= (\text{f9eae5f29b2815e31f11ac5d9c29fb01,37777748e56453377d5e262d90903f87}),$

$C_4 = 7\text{bcd1b0b73e32ba5b79cb140f2551504},$
$F[C_4] \dots F[C_1](K_1, K_2) =$
$= (\text{e980089683d00d4be37dd3434699b98f,f9eae5f29b2815e31f11ac5d9c29fb01}),$

$C_5 = 156\text{f6d791fab511deabb0c502fd18105},$
$F[C_5] \dots F[C_1](K_1, K_2) =$
$= (\text{b7bd70acea4460714f4ebe13835cf004,e980089683d00d4be37dd3434699b98f}),$

$C_6 = \text{a74af7efab73df160dd208608b9efe06},$
$F[C_6] \dots F[C_1](K_1, K_2) =$
$= (\text{1a46ea1cf6ccd236467287df93fdf974,b7bd70acea4460714f4ebe13835cf004}),$

$C_7 = \text{c9e8819dc73ba5ae50f5b570561a6a07},$
$F[C_7] \dots F[C_1](K_1, K_2) =$
$= (\text{3d4553d8e9cfec6815ebadc40a9ffd04,1a46ea1cf6ccd236467287df93fdf974}),$

$C_8 = \text{f6593616e6055689adfba18027aa2a08},$
$(K_3, K_4) = F[C_8] \dots F[C_1](K_1, K_2) =$
$= (\text{db31485315694343228d6aef8cc78c44,3d4553d8e9cfec6815ebadc40a9ffd04}).$

Итерационные ключи K_i имеют следующие значения:

$$K_1 = \text{8899aabbccddeeff0011223344556677},$$
$$K_2 = \text{fedcba98765432100123456789abcdef},$$
$$K_3 = \text{db31485315694343228d6aef8cc78c44},$$
$$K_4 = \text{3d4553d8e9cfec6815ebadc40a9ffd04},$$
$$K_5 = \text{57646468c44a5e28d3e59246f429f1ac},$$
$$K_6 = \text{bd079435165c6432b532e82834da581b},$$
$$K_7 = \text{51e640757e8745de705727265a0098b1},$$
$$K_8 = \text{5a7925017b9fdd3ed72a91a22286f984},$$
$$K_9 = \text{bb44e25378c73123a5f32f73cdb6e517},$$
$$K_{10} = \text{72e9dd7416bcf45b755dbaa88e4a4043}.$$

3.5 Контрольный пример для алгоритма зашифрования

В настоящем контрольном примере зашифрование производится при значениях итерационных ключей из п. 3.4. Пусть открытый текст, подлежащий зашифрованию, равен

$$a = 1122334455667700\text{ffeeddccbbaa9988},$$

тогда

$$X[K_1](a) = 99\text{bb}99\text{ff}99\text{bb}99\text{ffffffffffffffff},$$
$$SX[K_1](a) = \text{e87de8b6e87de8b6b6b6b6b6b6b6b6b6},$$
$$LSX[K_1](a) = \text{e297b686e355b0a1cf4a2f9249140830},$$
$$LSX[K_2]LSX[K_1](a) = \text{285e497a0862d596b36f4258a1c69072},$$
$$LSX[K_3]\dots LSX[K_1](a) = \text{0187a3a429b567841ad50d29207cc34e},$$
$$LSX[K_4]\dots LSX[K_1](a) = \text{ec9bdba057d4f4d77c5d70619dcad206},$$
$$LSX[K_5]\dots LSX[K_1](a) = \text{1357fd11de9257290c2a1473eb6bcde1},$$
$$LSX[K_6]\dots LSX[K_1](a) = \text{28ae31e7d4c2354261027ef0b32897df},$$
$$LSX[K_7]\dots LSX[K_1](a) = \text{07e223d56002c013d3f5e6f714b86d2d},$$
$$LSX[K_8]\dots LSX[K_1](a) = \text{cd8ef6cd97e0e092a8e4cca61b38bf65},$$
$$LSX[K_9]\dots LSX[K_1](a) = \text{0d8e40e4a800d06b2f1b37ea379ead8e}.$$

Результатом зашифрования является шифртекст

$$b = X[K_{10}]LSX[K_9]\dots LSX[K_1](a)=\text{7f679d90bebc24305a468d42b9d4edcd}.$$

3.6 Контрольный пример для алгоритма расшифрования

В настоящем контрольном примере расшифрование производится при значениях итерационных ключей из п. 3.4. Пусть шифртекст, подлежащий расшифрованию, равен шифртексту, полученному в предыдущем пункте:

$$b = \text{7f679d90bebc24305a468d42b9d4edcd},$$

тогда

$X[K_{10}](b) = 0\text{d8e40e4a800d06b2f1b37ea379ead8e},$

$L^{-1}X[K_{10}](b) = 8\text{a6b930a52211b45c5baa43ff8b91319},$

$S^{-1}L^{-1}X[K_{10}](b) = 76\text{ca149eef27d1b10d17e3d5d68e5a72},$

$S^{-1}L^{-1}X[K_9]S^{-1}L^{-1}X[K_{10}](b) = 5\text{d9b06d41b9d1d2d04df7755363e94a9},$

$S^{-1}L^{-1}X[K_8]\dots S^{-1}L^{-1}X[K_{10}](b) = 79487192\text{aa45709c115559d6e9280f6e},$

$S^{-1}L^{-1}X[K_7]\dots S^{-1}L^{-1}X[K_{10}](b) = \text{ae506924c8ce331bb918fc5bdfb195fa},$

$S^{-1}L^{-1}X[K_6]\dots S^{-1}L^{-1}X[K_{10}](b) = \text{bbffbfc8939eaaffafb8e22769e323aa},$

$S^{-1}L^{-1}X[K_5]\dots S^{-1}L^{-1}X[K_{10}](b) = 3\text{cc2f07cc07a8bec0f3ea0ed2ae33e4a},$

$S^{-1}L^{-1}X[K_4]\dots S^{-1}L^{-1}X[K_{10}](b) = \text{f36f01291d0b96d591e228b72d011c36},$

$S^{-1}L^{-1}X[K_3]\dots S^{-1}L^{-1}X[K_{10}](b) = 1\text{c4b0c1e950182b1ce696af5c0bfc5df},$

$S^{-1}L^{-1}X[K_2]\dots S^{-1}L^{-1}X[K_{10}](b) = 99\text{bb99ff99bb99ffffffffffffffff}.$

Результатом расшифрования является открытый текст

$$a = X[K_1]S^{-1}L^{-1}X[K_2]\dots S^{-1}L^{-1}X[K_{10}](b) =$$
$$=1122334455667700\text{ffeeddccbbaa9988}.$$