FEDERAL AGENCY
ON TECHNICAL REGULATION AND METROLOGY

NATIONAL STANDARD OF THE RUSSIAN FEDERATION

GOST R 34.12–2015

Information technology

CRYPTOGRAPHIC DATA SECURITY

Block ciphers

English Version

Moscow
Standartinform
2015

Foreword

1 DEVELOPED by the Center for Information Protection and Special Communications of the Federal Security Service of the Russian Federation with participation of the Open Joint-Stock company “Information Technologies and Communication Systems” (JSC “InfoTeCS”)

2 SUBMITTED by the Technical Committee for Standardization TC 26 “Cryptography and security mechanisms”

3 APPROVED AND PUT INTO EFFECT under Decree No 749-st of the Federal Agency on Technical Regulation and Metrology dated June 19, 2015

4 FIRST EDITION

Application of this Standard is ruled by GOST R 1.0-2012 (Section 8). Information on amendments to this Standard (as of January 1 of the current year) is published in National Standards annual information index. The text of revisions and amendments is published in National Standards monthly information indices. In case of revision (replacement) or cancellation of this Standard, a corresponding notification is published in National Standards monthly information index. Any corresponding information, notifications, and texts are also available in the public information system — on the official web-site of the Federal Agency on Technical Regulation and Metrology on the Internet (www.gost.ru).

Contents

Introduction

This Standard specifies block ciphers used in cryptographic methods of information protection.

The need for the development of this Standard was determined by the demand for block ciphers that support different block lengths and meet modern requirements for cryptographic strength and performance properties.

The terms and notions of this Standard comply with international standard ISO/IEC 10116 [1] and series of standards ISO/IEC 18033 [2], [3].

N o t e – The main part of this Standard is supplemented with Annex A.

NATIONAL STANDARD OF THE RUSSIAN FEDERATION

Information technology
CRYPTOGRAPHIC DATA SECURITY
Block ciphers

Effective date — 2016—01—01

1 Scope

This Standard specifies basic block ciphers used as cryptographic techniques for information processing and information protection including the provision of confidentiality, authenticity, and integrity of information during information transmission, processing and storage in computer-aided systems.

The cryptographic algorithms specified in this Standard are designed both for hardware and software implementation. They comply with modern cryptographic requirements, and put no restrictions on the confidentiality level of the protected information.

This Standard applies to developing, operation, and modernization of the information systems of various purposes.

2 Terms, definitions, and symbols

For the purposes of this Standard, the following terms and definitions apply.

2.1 Terms and definitions

2.1.1

encryption algorithm: process which transforms plaintext into ciphertext
[ISO/IEC 18033–1, clause 2.19]

Official Publication

2.1.2

decryption algorithm: process which transforms ciphertext into plaintext
[ISO/IEC 18033–1, clause 2.14]

2.1.3

basic block cipher: block cipher which for a given key provides one particular invertible mapping of the set of fixed-length plaintext blocks into ciphertext blocks of the same length

2.1.4

block: string of bits of a defined length
[ISO/IEC 18033–1, clause 2.6]

2.1.5

block cipher: symmetric cryptographic technique with the property that the encryption algorithm operates on a block of plaintext to yield a block of ciphertext
[ISO/IEC 18033–1, clause 2.7]

2.1.6

encryption: reversible transformation of data by a cipher to produce ciphertext from plaintext
[ISO/IEC 18033–1, clause 2.18]

2.1.7

round key: sequence of symbols derived from the key using the key schedule, and used to control the transformation in each round of an iterative block cipher

2.1.8

key: sequence of symbols that controls the operation of a cryptographic transformation
[ISO/IEC 18033–1, clause 2.21]

Note – In this Standard, keys in the form of sequences of bits are only considered.

2.1.9

plaintext: unencrypted information
[ISO/IEC 10116, clause 3.11]

2.1.10

key schedule: process which transforms the key into round keys

2.1.11

decryption: reversal of a corresponding encryption
[ISO/IEC 18033-1, clause 2.13]

2.1.12

symmetric cryptographic technique: cryptographic technique that uses the same key for both the originator's and recipient's transformation
[ISO/IEC 18033–1, clause 2.32]

2.1.13

cipher: cryptographic technique used to protect the confidentiality of data, and which consists of both encryption and decryption algorithms
[ISO/IEC 18033–1, clause 2.20]

2.1.14

ciphertext: data which has been transformed from plaintext to hide its information content
[ISO/IEC 10116, clause 3.3]

2.2 Symbols

For the purposes of this Standard, the following symbols apply.

V^*	the set of all binary vector-strings of finite length (hereinafter referred to as strings), including an empty string;
V_s	the set of all binary strings of length s, where s is a non-negative integer; substrings and string components are enumerated from right (lower order) to left (higher order) starting from zero;
$U \times W$	direct (Cartesian) product of two sets U and W;
$\|A\|$	the number of components (the length) of $A \in V^*$; if A is an empty string, then $\|A\| = 0$;
$A\|\|B$	concatenation of the strings A, $B \in V^*$, i.e. a string from $V_{\|A\| + \|B\|}$, where the substring with higher order components from $V_{\|A\|}$ is equal to A, and the substring with lower order components from $V_{\|B\|}$ is equal to B;
$A \lll_{11}$	cyclic shift of $A \in V_{32}$ by 11 positions in the direction of higher order components;
$\oplus$	bitwise addition modulo 2 of two binary strings of the same length;
$\mathbb{Z}_{2^s}$	the integer residue ring modulo 2^s;
$\boxplus$	the addition operation in $\mathbb{Z}_{2^{32}}$;
$\mathbb{F}$	the finite field $GF(2)[x]/p(x)$, where $p(x) = x^8 + x^7 + x^6 + x + 1 \in GF(2)[x]$; the elements of $\mathbb{F}$ are represented by integers; the integer $z_0 + 2 \cdot z_1 + \ldots + 2^7 \cdot z_7$, $z_i \in \{0, 1\}$, $i = 0, 1, \ldots, 7$, corresponds to the element $z_0 + z_1 \cdot \theta + \ldots + z_7 \cdot \theta^7 \in \mathbb{F}$, where θ is a residue class modulo $p(x)$ containing x.
$\mathrm{Vec}_s\colon \mathbb{Z}_{2^s} \to V_s$	the bijective mapping which for an element from $\mathbb{Z}_{2^s}$ puts into correspondence its binary representation, i.e. for any $z \in \mathbb{Z}_{2^s}$ represented as $z = z_0 + 2 \cdot z_1 + \ldots + 2^{s-1} \cdot z_{s-1}$, where $z_i \in \{0, 1\}$, $i = 0, 1, \ldots, s - 1$, the equality $\mathrm{Vec}_s(z) = z_{s-1}\|\|\ldots\|\|z_1\|\|z_0$ holds;
$\mathrm{Int}_s\colon V_s \to \mathbb{Z}_{2^s}$	the mapping inverse to the mapping Vec_s, i.e. $\mathrm{Int}_s = \mathrm{Vec}_s^{-1}$;

$\Delta: V_8 \to \mathbb{F}$ — the bijective mapping which maps a binary string from V_8 into an element of $\mathbb{F}$ as follows:
a string $z_7||\ldots||z_1||z_0$, $z_i \in \{0, 1\}$, $i = 0, 1, \ldots, 7$,
corresponds to the element $z_0 + z_1 \cdot \theta + \ldots + z_7 \cdot \theta^7 \in \mathbb{F}$;

$\nabla: \mathbb{F} \to V_8$ — the mapping inverse to the mapping Δ, i.e.$\nabla = \Delta^{-1}$;

$\Phi\Psi$ — a composition of mappings, where the mapping Ψ applies first;

Φ^s — a composition of mappings $\Phi^{s-1}$and Φ, where $\Phi^1 = \Phi$.

3 General provisions

This Standard specifies two basic block ciphers with block lengths of n = 128 bits and n = 64 bits.

N o t e . The cipher with block length of n = 128 bits, specified in this Standard, may be referred to as “Kuznyechik” block cipher.

N o t e . The cipher with block length of n = 64 bits, specified in this Standard, may be referred to as “Magma” block cipher.

4 128-bit Block cipher

4.1 Parameters

4.1.1 Bijective nonlinear mapping

The bijective nonlinear mapping is a substitution $\pi = \mathrm{Vec}_8\pi'\mathrm{Int}_8: V_8 \to V_8$, where $\pi': \mathbb{Z}_{2^8} \to \mathbb{Z}_{2^8}$. The values of π' are specified below as an array

$\pi' = (\pi'(0), \pi'(1), \ldots, \pi'(255))$:

π' = (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).

4.1.2 Linear mapping

The linear mapping is defined by a transformation $\ell\colon V_8^{16} \to V_8$ which is defined as follows:

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

for all $a_i \in V_8$, $i = 0, 1, \ldots, 15$, where the addition and multiplication operations are in the field $\mathbb{F}$, and the constants are field elements as defined above.

4.2 Transformations

The following transformations are used for encryption and decryption algorithms:

$X[k]\colon V_{128} \to V_{128}$ — $X[k](a) = k \oplus a$, where $k, a \in V_{128}$; (2)

$S\colon V_{128} \to V_{128}$ — $S(a) = S(a_{15}||\ldots||a_0) = \pi(a_{15})||\ldots||\pi(a_0)$, where $a = a_{15}||\ldots||a_0 \in V_{128}$, $a_i \in V_8$, $i = 0, 1, \ldots, 15$; (3)

$S^{-1}\colon V_{128} \to V_{128}$ — the inverse transformation to S, which may be calculated, for example, as follows: (4)
$S^{-1}(a) = S^{-1}(a_{15}||\ldots||a_0) = \pi^{-1}(a_{15})||\ldots||\pi^{-1}(a_0)$,
where $a = a_{15}||\ldots||a_0 \in V_{128}$, $a_i \in V_8$, $i = 0, 1, \ldots, 15$,
π^{-1} is the inverse to π;

$R\colon V_{128} \to V_{128}$ — $R(a) = R(a_{15}||\ldots||a_0) = \ell(a_{15}, \ldots, a_0)||a_{15}||\ldots||a_1$, where $a = a_{15}||\ldots||a_0 \in V_{128}$, $a_i \in V_8$, $i = 0, 1, \ldots, 15$; (5)

$L\colon V_{128} \to V_{128}$ — $L(a) = R^{16}(a)$, where $a \in V_{128}$; (6)

$R^{-1}\colon V_{128} \to V_{128}$ — the inverse transformation to R, which may be calculated, for example, as follows: (7)
$R^{-1}(a) = R^{-1}(a_{15}||\ldots||a_0) =$
$= a_{14}||a_{13}||\ldots||a_0||\ell(a_{14}, a_{13}, \ldots, a_0, a_{15})$,
where $a = a_{15}||\ldots||a_0 \in V_{128}$, $a_i \in V_8$, $i = 0, 1, \ldots, 15$;

$L^{-1}\colon V_{128} \to V_{128}$ — $L^{-1}(a) = (R^{-1})^{16}(a)$, where $a \in V_{128}$; (8)

$F[k]\colon V_{128} \times V_{128} \to V_{128} \times V_{128}$, — $F[k](a_1, a_0) = (LSX[k](a_1) \oplus a_0, a_1)$, where $k, a_0, a_1 \in V_{128}$. (9)

4.3 Key schedule

The key schedule uses round constants $C_i \in V_{128}$, $i = 1, 2, \ldots, 32$, defined as:

$$C_i = L(\mathrm{Vec}_{128}(i)),\ i = 1, 2, \ldots, 32. \quad (10)$$

Round keys $K_i \in V_{128}$, $i = 1, 2, \ldots, 10$, are derived from the key $K = k_{255}||\ldots||k_0 \in V_{256}$, $k_i \in V_1$, $i = 0, 1, \ldots, 255$, as follows:

$$K_1 = k_{255}||\ldots||k_{128};$$
$$K_2 = k_{127}||\ldots||k_0; \quad (11)$$
$$(K_{2i+1}, K_{2i+2}) = F[C_{8(i-1)+8}]\ldots F[C_{8(i-1)+1}](K_{2i-1}, K_{2i}),\ i = 1, 2, 3, 4.$$

4.4 Basic block cipher

4.4.1 Encryption algorithm

Depending on the values of the round keys $K_i \in V_{128}$, $i = 1, 2, \ldots, 10$ the encryption algorithm is a substitution $E_{K_1, \ldots, K_{10}}$ defined on V_{128} as follows:

$$E_{K_1, \ldots, K_{10}}(a) = X[K_{10}]LSX[K_9]\ldots LSX[K_2]LSX[K_1](a), \quad (12)$$

where $a \in V_{128}$.

4.4.2 Decryption algorithm

Depending on the values of the round keys $K_i \in V_{128}$, $i = 1, 2, \ldots, 10$ the decryption algorithm is a substitution $D_{K_1, \ldots, K_{10}}$ defined on V_{128} as follows:

$$D_{K_1, \ldots, K_{10}}(a) = X[K_1]S^{-1}L^{-1}X[K_2]\ldots S^{-1}L^{-1}X[K_9]S^{-1}L^{-1}X[K_{10}](a), \quad (13)$$

where $a \in V_{128}$.

5 64-bit Block cipher

5.1 Parameters

5.1.1 Bijective nonlinear mapping

The bijective nonlinear mapping is a substitution $\pi_i = \mathrm{Vec}_4\pi_i'\mathrm{Int}_4: V_4 \to V_4$, where $\pi_i': \mathbb{Z}_{2^4} \to \mathbb{Z}_{2^4}$, $i = 0, 1, \ldots, 7$. The values of π_i' are specified below as the following arrays:

$\pi_i' = (\pi_i'(0), \pi_i'(1), \ldots, \pi_i'(15))$, $i = 0, 1, \ldots, 7$:

π_0' = (12, 4, 6, 2, 10, 5, 11, 9, 14, 8, 13, 7, 0, 3, 15, 1);
π_1' = (6, 8, 2, 3, 9, 10, 5, 12, 1, 14, 4, 7, 11, 13, 0, 15);
π_2' = (11, 3, 5, 8, 2, 15, 10, 13, 14, 1, 7, 4, 12, 9, 6, 0);
π_3' = (12, 8, 2, 1, 13, 4, 15, 6, 7, 0, 10, 5, 3, 14, 9, 11);
π_4' = (7, 15, 5, 10, 8, 1, 6, 13, 0, 9, 3, 14, 11, 4, 2, 12);
π_5' = (5, 13, 15, 6, 9, 2, 12, 10, 11, 7, 8, 1, 4, 3, 14, 0);
π_6' = (8, 14, 2, 5, 6, 9, 1, 12, 15, 4, 11, 0, 13, 10, 3, 7);
π_7' = (1, 7, 14, 13, 0, 5, 8, 3, 4, 15, 10, 6, 9, 12, 11, 2).

5.2 Transformations

The following transformations are used for encryption and decryption algorithms:

$t: V_{32} \to V_{32}$ $\quad t(a) = t(a_7||\ldots||a_0) = \pi_7(a_7)||\ldots||\pi_0(a_0)$, where $a = a_7||\ldots||a_0 \in V_{32}$, $a_i \in V_4$, $i = 0, 1, \ldots, 7$; (14)

$g[k]: V_{32} \to V_{32}$ $\quad g[k](a) = (t(\mathrm{Vec}_{32}(\mathrm{Int}_{32}(a) \boxplus \mathrm{Int}_{32}(k)))) \lll_{11}$, where $k, a \in V_{32}$; (15)

$G[k]: V_{32} \times V_{32} \to V_{32} \times V_{32}$ $\quad G[k](a_1, a_0) = (a_0, g[k](a_0) \oplus a_1)$, where $k, a_0, a_1 \in V_{32}$; (16)

$G^*[k]: V_{32} \times V_{32} \to V_{64}$ $\quad G^*[k](a_1, a_0) = (g[k](a_0) \oplus a_1)||a_0$, where $k, a_0, a_1 \in V_{32}$. (17)

5.3 Key schedule

Round keys $K_i \in V_{32}$, $i = 1, 2, \ldots, 32$ are derived from the key $K = k_{255}||\ldots||k_0 \in V_{256}$, $k_i \in V_1$, $i = 0, 1, \ldots, 255$, as follows:

$$\begin{gathered} K_1 = k_{255}||\ldots||k_{224}; \\ K_2 = k_{223}||\ldots||k_{192}; \\ K_3 = k_{191}||\ldots||k_{160}; \\ K_4 = k_{159}||\ldots||k_{128}; \\ K_5 = k_{127}||\ldots||k_{96}; \\ K_6 = k_{95}||\ldots||k_{64}; \\ K_7 = k_{63}||\ldots||k_{32}; \\ K_8 = k_{31}||\ldots||k_0; \\ K_{i+8} = K_i,\ i = 1, 2, \ldots, 8; \\ K_{i+16} = K_i,\ i = 1, 2, \ldots, 8; \\ K_{i+24} = K_{9-i},\ i = 1, 2, \ldots, 8. \end{gathered} \qquad (18)$$

5.4 Basic block cipher

5.4.1 Encryption algorithm

Depending on the values of the round keys $K_i \in V_{32}$, $i = 1, 2, \ldots, 32$, the encryption algorithm is a substitution $E_{K_1, \ldots, K_{32}}$ defined on V_{64} as follows:

$$E_{K_1, \ldots, K_{32}}(a) = G^*[K_{32}]G[K_{31}]\ldots G[K_2]G[K_1](a_1, a_0), \qquad (19)$$

where $a = a_1||a_0 \in V_{64}$, $a_0, a_1 \in V_{32}$.

5.4.2 Decryption algorithm

Depending on the values of the round keys $K_i \in V_{32}$, $i = 1, 2, \ldots, 32$, the decryption algorithm is a substitution $D_{K_1, \ldots, K_{32}}$ defined on V_{64} as follows:

$$D_{K_1, \ldots, K_{32}}(a) = G^*[K_1]G[K_2]\ldots G[K_{31}]G[K_{32}](a_1, a_0), \qquad (20)$$

where $a = a_1 || a_0 \in V_{64}$, $a_0, a_1 \in V_{32}$.

Annex A

(informative)

Test examples

This Annex is for information only and is not a normative part of this Standard.

In this Annex, binary strings from V^*, whose length is a multiple of 4, are expressed in hexadecimal form, while the concatenation symbol ("||") is omitted. That is, a string $a \in V_{4n}$ shall be represented as

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

where $a_i \in \{0, 1, \dots, 9, a, b, c, d, e, f\}$, $i = 0, 1, \dots, n - 1$. The natural correspondence between binary strings of length 4 and hexadecimal strings of length 1 is given in Table 1. The transformation which for a binary string of length $4n$ puts into correspondence a hexadecimal string of length n and the inverse transformation are omitted for simplicity.

Table 1: Correspondence between binary strings and hexadecimal strings

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

A.1 128-bit Block cipher

A.1.1 Transformation S

S(ffeeddccbbaa99881122334455667700) = b66cd8887d38e8d77765aeea0c9a7efc,

S(b66cd8887d38e8d77765aeea0c9a7efc) = 559d8dd7bd06cbfe7e7b262523280d39,

S(559d8dd7bd06cbfe7e7b262523280d39) = 0c3322fed531e4630d80ef5c5a81c50b,

S(0c3322fed531e4630d80ef5c5a81c50b) = 23ae65633f842d29c5df529c13f5acda.

A.1.2 Transformation R

R(00000000000000000000000000000100) = 94000000000000000000000000000001,

R(94000000000000000000000000000001) = a5940000000000000000000000000000,

R(a5940000000000000000000000000000) = 64a59400000000000000000000000000,

R(64a59400000000000000000000000000) = 0d64a594000000000000000000000000.

A.1.3 Transformation L

L(64a59400000000000000000000000000) = d456584dd0e3e84cc3166e4b7fa2890d,

L(d456584dd0e3e84cc3166e4b7fa2890d) = 79d26221b87b584cd42fbc4ffea5de9a,

L(79d26221b87b584cd42fbc4ffea5de9a) = 0e93691a0cfc60408b7b68f66b513c13,

L(0e93691a0cfc60408b7b68f66b513c13) = e6a8094fee0aa204fd97bcb0b44b8580.

A.1.4 Key schedule

In this test example, the key is set to

K = 8899aabbccddeeff0011223344556677fedcba98765432100123456789abcdef.

K_1 = 8899aabbccddeeff0011223344556677,

K_2 = fedcba98765432100123456789abcdef.

C_1 = 6ea276726c487ab85d27bd10dd849401,

$X[C_1](K_1)$ = e63bdcc9a09594475d369f2399d1f276,

$SX[C_1](K_1)$ = 0998ca37a7947aabb78f4a5ae81b748a,

$LSX[C_1](K_1)$ = 3d0940999db75d6a9257071d5e6144a6,

$F[C_1](K_1, K_2) =$

= (c3d5fa01ebe36f7a9374427ad7ca8949, 8899aabbccddeeff0011223344556677).

C_2 = dc87ece4d890f4b3ba4eb92079cbeb02,

$F[C_2]F[C_1](K_1, K_2)$ =

= (37777748e56453377d5e262d90903f87, c3d5fa01ebe36f7a9374427ad7ca8949).

C_3 = b2259a96b4d88e0be7690430a44f7f03,

$F[C_3]\ldots F[C_1](K_1, K_2)$ =

= (f9eae5f29b2815e31f11ac5d9c29fb01, 37777748e56453377d5e262d90903f87).

C_4 = 7bcd1b0b73e32ba5b79cb140f2551504,

$F[C_4]\ldots F[C_1](K_1, K_2)$ =

= (e980089683d00d4be37dd3434699b98f, f9eae5f29b2815e31f11ac5d9c29fb01).

C_5 = 156f6d791fab511deabb0c502fd18105,

$F[C_5]\ldots F[C_1](K_1, K_2)$ =

= (b7bd70acea4460714f4ebe13835cf004, e980089683d00d4be37dd3434699b98f).

C_6 = a74af7efab73df160dd208608b9efe06,

$F[C_6]\ldots F[C_1](K_1, K_2)$ =

= (1a46ea1cf6ccd236467287df93fdf974, b7bd70acea4460714f4ebe13835cf004).

C_7 = c9e8819dc73ba5ae50f5b570561a6a07,

$F[C_7]\ldots F[C_1](K_1, K_2)$ =

= (3d4553d8e9cfec6815ebadc40a9ffd04, 1a46ea1cf6ccd236467287df93fdf974).

C_8 = f6593616e6055689adfba18027aa2a08,

$(K_3, K_4) = F[C_8]\ldots F[C_1](K_1, K_2)$ =

= (db31485315694343228d6aef8cc78c44, 3d4553d8e9cfec6815ebadc40a9ffd04).

The round keys K_i, i = 1, 2, …, 10, take the following values:

K_1 = 8899aabbccddeeff0011223344556677,

K_2 = fedcba98765432100123456789abcdef,

K_3 = db31485315694343228d6aef8cc78c44,

K_4 = 3d4553d8e9cfec6815ebadc40a9ffd04,

K_5 = 57646468c44a5e28d3e59246f429f1ac,

K_6 = bd079435165c6432b532e82834da581b,

K_7 = 51e640757e8745de705727265a0098b1,

K_8 = 5a7925017b9fdd3ed72a91a22286f984,

K_9 = bb44e25378c73123a5f32f73cdb6e517,

K_{10} = 72e9dd7416bcf45b755dbaa88e4a4043.

A.1.5 Encryption

In this test example, encryption is performed on the round keys specified in clause A.1.4. Let a plaintext be

a = 1122334455667700ffeeddccbbaa9988.

Then

$X[K_1](a)$ = 99bb99ff99bb99ffffffffffffffffff,

$SX[K_1](a)$ = e87de8b6e87de8b6b6b6b6b6b6b6b6b6,

$LSX[K_1](a)$ = e297b686e355b0a1cf4a2f9249140830,

$LSX[K_2]LSX[K_1](a)$ = 285e497a0862d596b36f4258a1c69072,

$LSX[K_3]\ldots LSX[K_1](a)$ = 0187a3a429b567841ad50d29207cc34e,

$LSX[K_4]\ldots LSX[K_1](a)$ = ec9bdba057d4f4d77c5d70619dcad206,

$LSX[K_5]\ldots LSX[K_1](a)$ = 1357fd11de9257290c2a1473eb6bcde1,

$LSX[K_6]\ldots LSX[K_1](a)$ = 28ae31e7d4c2354261027ef0b32897df,

$LSX[K_7]\ldots LSX[K_1](a)$ = 07e223d56002c013d3f5e6f714b86d2d,

$LSX[K_8]\ldots LSX[K_1](a)$ = cd8ef6cd97e0e092a8e4cca61b38bf65,

$LSX[K_9]\ldots LSX[K_1](a)$ = 0d8e40e4a800d06b2f1b37ea379ead8e.

The resulting ciphertext is

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

A.1.6 Decryption

In this test example, decryption is performed on the round keys specified in clause A.1.4. Let a ciphertext be equal to the one obtained in the previous clause:

b = 7f679d90bebc24305a468d42b9d4edcd.

Then

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

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

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

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

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

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

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

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

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

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

$S^{-1}L^{-1}X[K_2]\ldots S^{-1}L^{-1}X[K_{10}](b)$ = 99bb99ff99bb99ffffffffffffffffff.

The decrypted plaintext is

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

A.2 64-bit Block cipher

A.2.1 Transformation t

t(fdb97531) = 2a196f34,

t(2a196f34) = ebd9f03a,

t(ebd9f03a) = b039bb3d,

t(b039bb3d) = 68695433.

A.2.2 Transformation g

g[87654321](fedcba98) = fdcbc20c,

g[fdcbc20c](87654321) = 7e791a4b,

g[7e791a4b](fdcbc20c) = c76549ec,

g[c76549ec](7e791a4b) = 9791c849.

A.2.3 Key schedule

In this test example, the key is set to

K = ffeeddccbbaa99887766554433221100f0f1f2f3f4f5f6f7f8f9fafbfcfdfeff.

The round keys K_i, i = 1, 2, …, 32, take the following values:

K_1 = ffeeddcc,	K_9 = ffeeddcc,	K_{17} = ffeeddcc,	K_{25} = fcfdfeff,
K_2 = bbaa9988,	K_{10} = bbaa9988,	K_{18} = bbaa9988,	K_{26} = f8f9fafb,
K_3 = 77665544,	K_{11} = 77665544,	K_{19} = 77665544,	K_{27} = f4f5f6f7,
K_4 = 33221100,	K_{12} = 33221100,	K_{20} = 33221100,	K_{28} = f0f1f2f3,
K_5 = f0f1f2f3,	K_{13} = f0f1f2f3,	K_{21} = f0f1f2f3,	K_{29} = 33221100,
K_6 = f4f5f6f7,	K_{14} = f4f5f6f7,	K_{22} = f4f5f6f7,	K_{30} = 77665544,
K_7 = f8f9fafb,	K_{15} = f8f9fafb,	K_{23} = f8f9fafb,	K_{31} = bbaa9988,
K_8 = fcfdfeff,	K_{16} = fcfdfeff,	K_{24} = fcfdfeff,	K_{32} = ffeeddcc.

A.2.4 Encryption

In this test example, encryption is performed on the round keys specified in clause A.2.3. Let a plaintext be

a = fedcba9876543210.

Then

(a_1, a_0) = (fedcba98, 76543210),

$G[K_1](a_1, a_0)$ = (76543210, 28da3b14),

$G[K_2]G[K_1](a_1, a_0)$ = (28da3b14, b14337a5),

$G[K_3]\ldots G[K_1](a_1, a_0)$ = (b14337a5, 633a7c68),

$G[K_4]\ldots G[K_1](a_1, a_0)$ = (633a7c68, ea89c02c),

$G[K_5]\ldots G[K_1](a_1, a_0)$ = (ea89c02c, 11fe726d),

$G[K_6]\ldots G[K_1](a_1, a_0)$ = (11fe726d, ad0310a4),

$G[K_7]\ldots G[K_1](a_1, a_0)$ = (ad0310a4, 37d97f25),

$G[K_8]\ldots G[K_1](a_1, a_0)$ = (37d97f25, 46324615),

$G[K_9]\ldots G[K_1](a_1, a_0)$ = (46324615, ce995f2a),

$G[K_{10}]\ldots G[K_1](a_1, a_0)$ = (ce995f2a, 93c1f449),

$G[K_{11}]\ldots G[K_1](a_1, a_0)$ = (93c1f449, 4811c7ad),

$G[K_{12}]\ldots G[K_1](a_1, a_0)$ = (4811c7ad, c4b3edca),

$G[K_{13}]\ldots G[K_1](a_1, a_0)$ = (c4b3edca, 44ca5ce1),

$G[K_{14}]\ldots G[K_1](a_1, a_0)$ = (44ca5ce1, fef51b68),

$G[K_{15}]\ldots G[K_1](a_1, a_0)$ = (fef51b68, 2098cd86),

$G[K_{16}]\ldots G[K_1](a_1, a_0)$ = (2098cd86, 4f15b0bb),

$G[K_{17}]\ldots G[K_1](a_1, a_0)$ = (4f15b0bb, e32805bc),

$G[K_{18}]\ldots G[K_1](a_1, a_0)$ = (e32805bc, e7116722),

$G[K_{19}]\ldots G[K_1](a_1, a_0)$ = (e7116722, 89cadf21),

$G[K_{20}]\ldots G[K_1](a_1, a_0)$ = (89cadf21, bac8444d),

$G[K_{21}]\ldots G[K_1](a_1, a_0)$ = (bac8444d, 11263a21),

$G[K_{22}]\ldots G[K_1](a_1, a_0)$ = (11263a21, 625434c3),

$G[K_{23}]\ldots G[K_1](a_1, a_0)$ = (625434c3, 8025c0a5),

$G[K_{24}]\ldots G[K_1](a_1, a_0)$ = (8025c0a5, b0d66514),

$G[K_{25}]\ldots G[K_1](a_1, a_0)$ = (b0d66514, 47b1d5f4),

$G[K_{26}]\ldots G[K_1](a_1, a_0)$ = (47b1d5f4, c78e6d50),

$G[K_{27}]\ldots G[K_1](a_1, a_0)$ = (c78e6d50, 80251e99),

$G[K_{28}]\ldots G[K_1](a_1, a_0)$ = (80251e99, 2b96eca6),

$G[K_{29}]\ldots G[K_1](a_1, a_0)$ = (2b96eca6, 05ef4401),

$G[K_{30}]\ldots G[K_1](a_1, a_0)$ = (05ef4401, 239a4577),

$G[K_{31}]\ldots G[K_1](a_1, a_0)$ = (239a4577, c2d8ca3d).

The resulting ciphertext is

$$b = G^*[K_{32}]G[K_{31}]\ldots G[K_1](a_1, a_0) = \text{4ee901e5c2d8ca3d}.$$

A.2.5 Decryption

In this test example, decryption is performed on the round keys specified in clause A.2.3. Let a ciphertext be equal to the one obtained in the previous clause:

$$b = 4ee901e5c2d8ca3d.$$

Then

(b_1, b_0) = (4ee901e5, c2d8ca3d),

$G[K_{32}](b_1, b_0)$ = (c2d8ca3d, 239a4577),

$G[K_{31}]G[K_{32}](b_1, b_0)$ = (239a4577, 05ef4401),

$G[K_{30}]\ldots G[K_{32}](b_1, b_0)$ = (05ef4401, 2b96eca6),

$G[K_{29}]\ldots G[K_{32}](b_1, b_0)$ = (2b96eca6, 80251e99),

$G[K_{28}]\ldots G[K_{32}](b_1, b_0)$ = (80251e99, c78e6d50),

$G[K_{27}]\ldots G[K_{32}](b_1, b_0)$ = (c78e6d50, 47b1d5f4),

$G[K_{26}] \ldots G[K_{32}](b_1, b_0)$ = (47b1d5f4, b0d66514),

$G[K_{25}]\ldots G[K_{32}](b_1, b_0)$ = (b0d66514, 8025c0a5),

$G[K_{24}]\ldots G[K_{32}](b_1, b_0)$ = (8025c0a5, 625434c3),

$G[K_{23}]\ldots G[K_{32}](b_1, b_0)$ = (625434c3, 11263a21),

$G[K_{22}]\ldots G[K_{32}](b_1, b_0)$ = (11263a21, bac8444d),

$G[K_{21}]\ldots G[K_{32}](b_1, b_0)$ = (bac8444d, 89cadf21),

$G[K_{20}]\ldots G[K_{32}](b_1, b_0)$ = (89cadf21, e7116722),

$G[K_{19}]\ldots G[K_{32}](b_1, b_0)$ = (e7116722, e32805bc),

$G[K_{18}]\ldots G[K_{32}](b_1, b_0)$ = (e32805bc, 4f15b0bb),

$G[K_{17}]\ldots G[K_{32}](b_1, b_0)$ = (4f15b0bb, 2098cd86),

$G[K_{16}]\ldots G[K_{32}](b_1, b_0)$ = (2098cd86, fef51b68),

$G[K_{15}]\ldots G[K_{32}](b_1, b_0)$ = (fef51b68, 44ca5ce1),

$G[K_{14}]\ldots G[K_{32}](b_1, b_0)$ = (44ca5ce1, c4b3edca),

$G[K_{13}]\ldots G[K_{32}](b_1, b_0)$ = (c4b3edca, 4811c7ad),

$G[K_{12}]\ldots G[K_{32}](b_1, b_0)$ = (4811c7ad, 93c1f449),

$G[K_{11}]\ldots G[K_{32}](b_1, b_0)$ = (93c1f449, ce995f2a),

$G[K_{10}]\ldots G[K_{32}](b_1, b_0)$ = (ce995f2a, 46324615),

$G[K_9]\ldots G[K_{32}](b_1, b_0)$ = (46324615, 37d97f25),

$G[K_8]\ldots G[K_{32}](b_1, b_0)$ = (37d97f25, ad0310a4),

$G[K_7]\ldots G[K_{32}](b_1, b_0)$ = (ad0310a4, 11fe726d),

$G[K_6]\ldots G[K_{32}](b_1, b_0)$ = (11fe726d, ea89c02c),

$G[K_5]\ldots G[K_{32}](b_1, b_0)$ = (ea89c02c, 633a7c68),

$G[K_4]\ldots G[K_{32}](b_1, b_0)$ = (633a7c68, b14337a5),

$G[K_3]\ldots G[K_{32}](b_1, b_0)$ = (b14337a5, 28da3b14),

$G[K_2]\ldots G[K_{32}](b_1, b_0)$ = (28da3b14, 76543210).

The decrypted plaintext is

$$a = G^*[K_1]G[K_2]\ldots G[K_{32}](b_1, b_0) = \text{fedcba9876543210}.$$

Bibliography

[1] ISO/IEC 10116:2006	Information technology — Security techniques — Modes of operation for an *n*-bit block cipher
[2] ISO/IEC 18033-1:2015	Information technology — Security techniques Encryption algorithms — Part 1: General
[3] ISO/IEC 18033-3:2010	Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers

[*] These International ISO/IEC standards are available at the FSUE *"Standartinform*" of the Federal Agency on Technical Regulation and Metrology.

UDC 681.3.06:006.354 OKS 35. 040 OKSTU 5002 P 85

Keywords: information technology, cryptographic data security, symmetric cryptographic technique, encryption, decryption, block cipher, key