Cryptographic Protection of Information Through Bloc Encryption

Standartinform, Moscow, Russian Federation

FEDERAL AGENCY
Technical and Metrology Regulations
GOST National Standards of Russian Federation

English Version: Eric Filiol
This standard is still a project and is not approved yet

1 Introduction

This document presents the Russian Federation project for a new bloc encryption standard. It is a preliminary version which has not been validated or approved by the Russian Federation yet. The original document (in Russian) from which the present translation has been performed is available in [1].

2 Notation

This project of standard uses the following symbols and notation:

V^*	set of binary vectors y having a bounded length (including the void string)
V_s	set of all binary strings of length s, where s denotes a non negative integer (bits are written from the right to the left starting from index 0)
$U \times W$	direct sum (Cartesian product) of sets U and W
$\|A\|$	number of components (length) of the string $A \in V^*$ (if A is the null string, then $\|A\| = 0$)
$A\|\|B$	concatenation of strings $A, B \in V^*$, as element from $V_{\|A\|+\|B\|}$, in which the left substring of $V_{\|A\|}$ coincide with the string A, and the right substring of $V_{\|B\|}$ coincide with the string B
$A <<< 11$	cyclic shift of string A of 11 positions to the left (toward the most significant bit)
$\oplus$	bitwise addition modulo 2 (xor) of two strings of same length
$\mathbb{Z}_{2^s}$	ring of integers modulo 2^s
$\boxplus$	addition operator in ring $\mathbb{Z}_{2^{32}}$
$\mathbb{F}$	finite field $GF(2)[X]/p(x)$, where $p(x) = x^8 \oplus x^7 \oplus x^6 \oplus x \oplus 1 \in GF(2)[x]$; elements of finite field $\mathbb{F}$ are represented by integers having the form $z_0 + z_1.\theta + ... + z_7.\theta^7 \in \mathbb{F}$, where $z_i \in \{0, 1\}, i = 0, 1...7$, and θ represents the class residu modulo $p(x)$, containing x; it corresponds to the integer $z_0 + 2.z_1 + ... + 2^7.z7$

$Vec_s : Z_{2^s} \to V_s$	bijection which maps its binary representation to any element in ring $\mathbb{Z}_{2^s}$, that is to say, for any $z \in \mathbb{Z}_{2^s}$, described as $z = z_0 + 2.z_1 + ... + 2^{s-1}.z_{s-1}$, where $z_i \in \{0,1\}, i = 0,1...s-1$, its binary representation is $Vec_s(z) = z_{s-1}\|\|...\|\|z_1\|\|z_0$
$Int_s : V_s \to Z_{2^s}$	inverse bijection of Vec_s, that is to say $Int_s = Vec_s^{-1}$
$\triangle : V_8 \to \mathbb{F}$	bijection which maps a binary string of V_8 to an element of $\mathbb{F}$ as follows: to string $z_7\|\|...\|\|z_1\|\|z_0, z_i \in \{0,1\}, i = 0,1...7$ corresponds the element $z_0 + z_1.\theta + ... + z_7.\theta^7 \in \mathbb{F}$
$\nabla : \mathbb{F} \to V_8$	inverse bijection of $\triangle$, that is to say $\nabla = \triangle^{-1}$
$\Phi\Psi$	composition of functions in which function Ψ is applied first
Φ^s	iteration of composition of Φ^{s-1} with Φ, where $\Phi^1 = \Phi$

3 General Conditions

This project of standard described two block encryption algorithms having bloc length equal to $n = 128$ and $n = 64$ bits respectively.

In the present dcument, the block encryption algorithm standard with block length $n = 128$ bits is called "*Grasshopper*" ("*Kuznyechik*") algorithm.

In the present dcument, the block encryption algorithm standard with block length $n = 64$ bits (present day standard which is given here for historical continuity[1]) can be referred to "*GOST 28147-89*" algorithm.

4 Description of the *Grasshopper* Algorithm (block length $n = 128$ bits)

4.1 Parameter Values

Nonlinear bijective transformation.- A nonlinear bijective transformation applies a permutation $Vec_8\pi' Int_8 : V_8 \to V_8$ where $\pi' : \mathbb{Z}_{2^s} \to \mathbb{Z}_{2^s}$.

Permutation values π', are given as an array $\pi' = (\pi'(0), \pi'(1), \ldots, \pi'(255))$:

[1] This part is not translated yet and will be soon.

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)

Linear transformation.- This linear transformation is described by bijection $\ell = V_8^{16} \rightarrow V_8$, which is defined as follows:

$$\ell(a_{15}, ..., a_0) = \nabla(148.\triangle(a_{15}) + 32.\triangle(a_{14}) + 133.\triangle(a13) + 16.\triangle(a_{12}) + 194.\triangle(a_{11}) + 192.\triangle(a_{10}) + 1.\triangle(a_9) + 251.\triangle(a_8) + 1.\triangle(a_7) + 192.\triangle(a_6) + 194.\triangle(a_5) + 16.\triangle(a_4) + 133.\triangle(a_3) + 32.\triangle(a_2) + 148.\triangle(a_1) + 1.\triangle(a_0))$$

for all $a_i \in V_8, i = 0, 1, ..., 15$, where addition and multiplication operations are performed in $\mathbb{F}$.

4.2 Conversion

Encryption and decryption algorithms use the following conversion functions:

$X[k] : V_{128} \rightarrow V_{128}$	$X[k](a) = k \oplus a$ where $k, a \in V_{128}$
$S : V_{128} \rightarrow V_{128}$	$S(a) = S(a_{15}\|\|...\|\|a_0) = \pi(a_{a_{15}})\|\|...\|\|\pi(a_0)$, where $a = a_{15}\|\|...\|\|a_0 \in V_128, a_i \in V_8, i = 0, 1, ..., 15$
$S^{-1} : V_{128} \rightarrow V_{128}$	inverse conversion of S which can be computed as follows: $S^{-1}(a) = S^{-1}(a_{15}\|\|...\|\|a_0) = \pi^{-1}(a_{15})\|\|...\|\|\pi^{-1}(a_0)$ where $a = a_{15}\|\|...\|\|a_0 \in V128, a_i \in V_8, i = 0, 1, ..., 15$ and where π^{-1} describes the inverse substitution of permutation π

$R:\ V_{128} \to V_{128}$	$R(a) = R(a_{15}\|\|...\|\|a_0) = \ell(a_{15},...,a_0)\|\|a_{15}\|\|...\|\|a_1$, where $a = a_{15}\|\|...\|\|a_0 \in V_{128}, a_i \in V_8, i = 0,1,...,15$
$L:\ V_{128} \to V_{128}$	$L(a) = R^{16}(a)$ where $a \in V_{128}$
$R^{-1}:\ V_{128} \to V_{128}$	inverse transformation of transform R, which can be computed as follows: $R^{-1}(a) = R^{-1}(a_{15}\|\|...\|\|a_0) = a_{14}\|\|a_{13}\|\|...\|\|a0\|\|\ell(a_{14},a_{13},...,a_0,a_{15})$, where $a = a_{15}\|\|...\|\|a_0 \in V_{128}, a_i \in V_8, i = 0,1,...,15$
$L^{-1}:\ V_{128} \to V_{128}$	$L^{-1}(a) = (R^{-1})^{16}(a)$ where $a \in V_{128}$
$F[k]:\ V_{128} \times V_{128} \to V_{128} \times V_{128}$	$F[k](a_1,a_0) = (LSX[k](a_1) \oplus a_0, a_1)$ where $k, a_0, a_1 \in V_{128}$

4.3 (Sub)Key scheduling

The algorithm uses subkeys $C_i \in V_{128}, i = 1,2,...,32$ which are defined as follows:

$$C_i = L(Vec_{128}(i)), \qquad i = 1,2,...,32$$

Subkeys $K_i \in V_{128},\ i = 1,2,...,10$ are produced by an iterative process from a master key $K_0 = k_{255}||...||k_0 \in V_{256}, k_i \in V_1,\ i = 0,1,...,255$ according to the following equations:

$$K_1 = k_{255}||...||k_{128}$$

$$K_2 = k_{127}||...||k_0$$

$$\cdots$$

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

4.4 Description of the encryption algorithm

Encryption algorithm.- The encryption with subkeys $K_i \in V_{128}, i = 1,2,...,10$ uses the substitution $E_{K_1,...,K_{10}}$ which is defined on the set V_{128} according to equation:

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

where $a \in V_{128}$.

Decryption algorithm.- The decryption with subkeys $K_i \in V_{128}, i = 1, 2, ..., 10$ uses the substitution $D_{K_1,...,K_{10}}$ which is defined on the set V_{128} according to equation:

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

where $a \in V_{128}$.

A *Grasshopper* Algorithm Test Vectors (block length $n = 128$ bits)

This appendix is given for reference and validation purposes but it is not part of the standard. In this section, binary strings in V^* (whose length is a multiple of 4) are written in hexadecimal and the concatenation operator ("$||$") is omitted. In other word, the vector $a \in V_{4n}$ is represented by

$$a_{n-1}a_{n-2}\dots a_1a_0$$

where $a_i \in \{0, 1, ..., 9, A, B, C, D, E, F\}, i = 0, 1, ..., n-1$.

A.1 Encryption Algorithm (block length $n = 128$ bits)

Conversion S

$$S(ffeeddccbbaa99881122334455667700) = b66cd8887d38e8d77765aeea0c9a7efc,$$
$$S(b66cd8887d38e8d77765aeea0c9a7efc) = 559d8dd7bd06cbfe7e7b262523280d39,$$
$$S(559d8dd7bd06cbfe7e7b262523280d39) = 0c3322fed531e4630d80ef5c5a81c50b,$$
$$S(0c3322fed531e4630d80ef5c5a81c50b) = 23ae65633f842d29c5df529c13f5acda.$$

Conversion R

$$R(00000000000000000000000000000100) = 94000000000000000000000000000001,$$
$$R(94000000000000000000000000000001) = a5940000000000000000000000000000,$$
$$R(a5940000000000000000000000000000) = 64a59400000000000000000000000000,$$
$$R(64a59400000000000000000000000000) = 0d64a594000000000000000000000000.$$

Conversion L

$$L(64a59400000000000000000000000000) = d456584dd0e3e84cc3166e4b7fa2890d,$$
$$L(d456584dd0e3e84cc3166e4b7fa2890d) = 79d26221b87b584cd42fbc4ffea5de9a,$$
$$L(79d26221b87b584cd42fbc4ffea5de9a) = 0e93691a0cfc60408b7b68f66b513c13,$$
$$L(0e93691a0cfc60408b7b68f66b513c13) = e6a8094fee0aa204fd97bcb0b44b8580.$$

(Sub)Key scheduling.- In the present test vectors set, the master key has value:

$$K = 8899aabbccddeeff0011223344556677fedcba98765432100123456789abcdef$$

From this master key, we have:

$$\begin{aligned}
K_1 &= \text{8899aabbccddeeff0011223344556677}\\
K_2 &= \text{fedcba98765432100123456789abcdef}
\end{aligned}$$

$$\begin{aligned}
C_1 &= \text{6ea276726c487ab85d27bd10dd849401}\\
X[C_1](K_1) &= \text{e63bdcc9a09594475d369f2399d1f276}\\
SX[C_1](K_1) &= \text{0998ca37a7947aabb78f4a5ae81b748a}\\
LSX[C_1](K_1) &= \text{3d0940999db75d6a9257071d5e6144a6}\\
F[C_1](K_1, K_2) &= (\text{C3d5fa01ebe36f7a9374427ad7ca8949},\\
&\quad \text{8899aabbccddeeff0011223344556677})
\end{aligned}$$

$$\begin{aligned}
C_2 &= \text{dc87ece4d890f4b3ba4eb92079cbeb02}\\
F[C_2]F[C_1](K_1, K_2) &= (\text{37777748e56453377d5e262d90903f87},\\
&\quad \text{c3d5fa01ebe36f7a9374427ad7ca8949})\\
C_3 &= \text{b2259a96b4d88e0be7690430a44f7f03}\\
F[C_3]...F[C_1](K_1, K_2) &= (\text{F9eae5f29b2815e31f11ac5d9c29fb01},\\
&\quad \text{37777748e56453377d5e262d90903f87})\\
C_4 &= \text{7bcd1b0b73e32ba5b79cb140f2551504}\\
F[C_4]...F[C_1](K_1, K_2) &= (\text{E980089683d00d4be37dd3434699b98f},\\
&\quad \text{f9eae5f29b2815e31f11ac5d9c29fb01})\\
C_5 &= \text{156f6d791fab511deabb0c502fd18105}\\
F[C_5]...F[C_1](K_1, K_2) &= (\text{B7bd70acea4460714f4ebe13835cf004},\\
&\quad \text{e980089683d00d4be37dd3434699b98f})\\
C_6 &= \text{a74af7efab73df160dd208608b9efe06}\\
F[C_6]...F[C_1](K_1, K_2) &= (\text{1a46ea1cf6ccd236467287df93fdf974},\\
&\quad \text{b7bd70acea4460714f4ebe13835cf004})\\
C_7 &= \text{c9e8819dc73ba5ae50f5b570561a6a07}\\
F[C_7]...F[C_1](K_1, K_2) &= (\text{3d4553d8e9cfec6815ebadc40a9ffd04},\\
&\quad \text{1a46ea1cf6ccd236467287df93fdf974})\\
C_8 &= \text{f6593616e6055689adfba18027aa2a08}\\
(K_3, K_4) = F[C_8]...F[C_1](K_1, K_2) &= (\text{Db31485315694343228d6aef8cc78c44},\\
&\quad \text{3d4553d8e9cfec6815ebadc40a9ffd04})
\end{aligned}$$

Subkeys $K_i, i = 1, 2, ..., 10$ takes then the following values:

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

Encryption algorithm.- For the present test vectors set, encryption is performed with the subkey values given in the previous Subsection. Let us consider the encryption of the plaintext block

$$a = 1122334455667700ffeeddccbbaa9988$$

We then obtain:

$$
\begin{aligned}
X[K_1](a) &= 99bb99ff99bb99ffffffffffffffffff \\
SX[K_1](a) &= e87de8b6e87de8b6b6b6b6b6b6b6b6b6 \\
LSX[K_1](a) &= e297b686e355b0a1cf4a2f9249140830 \\
LSX[K_2]LSX[K_1](A) &= 285e497a0862d596b36f4258a1c69072 \\
LSX[K_3]...LSX[K_1](a) &= 0187a3a429b567841ad50d29207cc34e \\
LSX[K_4]...LSX[K_1](a) &= ec9bdba057d4f4d77c5d70619dcad206 \\
LSX[K_5]...LSX[K_1](A) &= 1357fd11de9257290c2a1473eb6bcde1 \\
LSX[K_6]...LSX[K_1](a) &= 28ae31e7d4c2354261027ef0b32897df \\
LSX[K_7]...LSX[K_1](a) &= 07e223d56002c013d3f5e6f714b86d2d \\
LSX[K_8]...LSX[K_1](a) &= cd8ef6cd97e0e092a8e4cca61b38bf65 \\
LSX[K_9]...LSX[K_1](a) &= 0d8e40e4a800d06b2f1b37ea379ead8e
\end{aligned}
$$

The last result is encrypted to produce the ciphertext bloc as follows

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

Decryption algorithm.- For the present test vectors set, encryption is performed with the subkey values given in the previous Subsection. Let us consider the ciphertext block obtained previously:

$$b = 7f679d90bebc24305a468d42b9d4edcd$$

We then obtain:

$$\begin{aligned}
X[K_{10}](b) &= 0d8e40e4a800d06b2f1b37ea379ead8e \\
L^{-1}X[K_{10}](b) &= 8a6b930a52211b45c5baa43ff8b91319 \\
S^{-1}L^{-1}X[K_{10}](b) &= 76ca149eef27d1b10d17e3d5d68e5a72 \\
S^{-1}L^{-1}X[K_9]S^{-1}L^{-1}X[K_{10}](b) &= 5d9b06d41b9d1d2d04df7755363e94a9 \\
S^{-1}L^{-1}X[K_8]...S^{-1}L^{-1}X[K_{10}](b) &= 79487192aa45709c115559d6e9280f6e \\
S^{-1}L^{-1}X[K_7]...S^{-1}L^{-1}X[K_{10}](b) &= ae506924c8ce331bb918fc5bdfb195fa \\
S^{-1}L^{-1}X[K_6]...S^{-1}L^{-1}X[K_{10}](b) &= bbffbfc8939eaaffafb8e22769e323aa \\
S^{-1}L^{-1}X[K_5]...S^{-1}L^{-1}X[K_{10}](b) &= 3cc2f07cc07a8bec0f3ea0ed2ae33e4a \\
S^{-1}L^{-1}X[K_4]...S^{-1}L^{-1}X[K_{10}](b) &= f36f01291d0b96d591e228b72d011c36 \\
S^{-1}L^{-1}X[K_3]...S^{-1}L^{-1}X[K_{10}](b) &= 1c4b0c1e950182b1ce696af5c0bfc5df \\
S^{-1}L^{-1}X[K_2]...S^{-1}L^{-1}X[K_{10}](b) &= 99bb99ff99bb99ffffffffffffffffff
\end{aligned}$$

The last result produces the resulting plaintext block

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

References

1. Standardization Technical Committee for "Cryptographic Protection of Information" (2013). `http://www.tc26.ru/standard/draft/GOSTR-bsh.pdf` (in Russian).