Low-Weight and Hi-End: Draft Russian Encryption Standard

Vasily Shishkin, Denis Dygin, Ivan Lavrikov,
Grigory Marshalko, Vladimir Rudskoy, Dmitry Trifonov

Abstract

We give a brief analysis for the current demands in standardized encryption algorithms and present a draft of a new block cipher which in the nearest future would ”keep company with GOST 28147-89” in a new Russian encryption standard.

Keywords: GOST 28147-89, block cipher, encryption standard.

1 Present and future

The present state-of-the-art in practical information security is highlighted by the extreme diversity of possible applications of cryptographic mechanisms: from restricted environments (like RFID-tags and smart cards) to protection of broadband Internet data channels. Each of the possible implementation areas impose specific requirements on implementation specifications of cryptographic mechanisms, which sometime contradict with the desired cryptographic features.

The question is: whether we should try to make an all-in-one ”Swiss army knife” which would actually be a compromise between implementation and cryptographic requirements, or develop a set of primitives, each for the possible application area.

The risk of the first approach is a possibly narrow security margin of a cryptographic primitive. Adoption of the second approach implies that we should maintain a reasonable diversity of cryptographic primitives, but shouldn’t make it to large.

GOST 28147-89 [1] is the only cipher standardized in the Russian Federation now. It was published in the end of 80's and since that time is widely used all over the world. It has been thoroughly studied by Russian and foreign cryptanalysts and has proven its cryptographic strength.

The only theoretical attacks for the full version of GOST 28147-89 were published a few years ago [3, 4]. They do not threaten the security of the cipher since the obtained time complexity for these attacks is considered greater than enough for providing security in foreseeable future, and data complexity exceed the maximum amount of data that could be encrypted by a block cipher [5]. It's interesting to mention that by a simple change of the key schedule these attacks could be made inapplicable [6].

At the same time, as it is shown by Poschmann et al. in [2], GOST 28147-89 is an extremely lightweight-friendly cipher and has the complexity of hardware implementation comparable with the complexities of implementations of algorithms with significantly shorter key lengths.

However from a point of view of encrypting large amounts of data 64-bit block length is insufficient. So, despite the fact that GOST 28147-89 is secure and has efficient implementation, there is a need for a new block cipher with larger block length. In this article we present a new Russian draft encryption standard with 128-bit block length and 256-bit key length, which in the nearest future would be also standardized in the Russian Federation (see also [7]).

New algorithm is based on well known and stabilized principles following current state-of-the-art in block cipher design strategies. Our main concern was to built software oriented cipher with as large security margin as possible.

2 Description of a draft encryption standard

2.1 Definitions and notation

In the rest of the article the following notations are used:

V_n the set of all binary strings of the length n, where $n -$ is a natural number. Sub-strings numbering goes from the right to the left starting from zero;

$A \| B$ a concatenation of strings $A, B \in V^*$, i.e. a string from the set $V_{|A|+|B|}$, where the left substring is equal to A, and the right string is equal to B;

$\oplus$ bit-wise addition modulo 2;

Z_m the set $\{0, 1, \ldots, m-1\}$;

P a finite field $GF(2)[x]/g(x)$, where $g(x) = x^8 + x^7 + x^6 + x + 1 \in GF(2)[x]$. Elements of P could be represented as integer numbers. The number $b_0 + b_1 \cdot 2 + ... + b_7 \cdot 2^7 \in Z_{2^8}$, $b_i \in \{0, 1\}$, $i = 0, 1, \ldots, 7$, corresponds to the element $b_0 + b_1 \cdot \theta + ... + b_7 \cdot \theta^7 \in P$, where θ denotes a residue class modulo $g(x)$, which contains x;

$\mathcal{V}_n : Z_{2^n} \to V_n$ a bijective mapping, which maps each element z from Z_{2^n}, where $z = z_0 + 2 \cdot z_1 + \ldots + 2^{n-1} \cdot z_{n-1}$, $z_j \in \{0, 1\}$, $j = 0, 1, \ldots, n-1$, to his binary representation, i. e. the following equation holds: $\mathcal{V}_n(z) = z_{n-1} \| \ldots \| z_1 \| z_0$;

$\mathcal{I}_n : V_n \to Z_{2^n}$ an inverse mapping to $\mathcal{V}_n$, i. e. $\mathcal{I}_n = \mathcal{V}_n^{-1}$;

$\Delta : V_8 \to P$ a bijective mapping, which maps a binary string from V_8 to an element of the field P as follows: the string $b_7 || b_6 || \ldots || b_0$, $b_i \in V_1$, $i = 0, 1, \ldots, 7$, corresponds to the element $b_0 + b_1 \cdot \theta + \ldots + b_7 \cdot \theta^7 \in P$;

$\nabla : P \to V_8$ an inverse mapping to Δ, i.e. $\nabla = \Delta^{-1}$;

$\Phi\Psi$ a composition of mappings, where Ψ is the first to operate.

2.2 Parameters

2.2.1 Nonlinear bijective mapping

The nonlinear mapping of the algorithm is represented by the substitution $\pi = \mathcal{V}_8 \pi' \mathcal{I}_8 : V_8 \to V_8$, where $\pi' : Z_{2^8} \to Z_{2^8}$ is presented below as $\pi' = (\pi'(0), \pi'(1), \ldots, \pi'(255))$:

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).

2.2.2 Linear mapping

The linear mapping is denoted by $l : V_8^{16} \to V_8$ and defined as:

$l(a_{15}, \ldots, a_0) = \nabla(148 \cdot \Delta(a_{15}) + 32 \cdot \Delta(a_{14}) + 133 \cdot \Delta(a_{13}) + 16 \cdot \Delta(a_{12}) + 194 \cdot \Delta(a_{11}) + 192 \cdot \Delta(a_{10}) + 1 \cdot \Delta(a_9) + 251 \cdot \Delta(a_8) + 1 \cdot \Delta(a_7) + 192 \cdot \Delta(a_6) + 194 \cdot \Delta(a_5) + 16 \cdot \Delta(a_4) + 133 \cdot \Delta(a_3) + 32 \cdot \Delta(a_2) + 148 \cdot \Delta(a_1) + 1 \cdot \Delta(a_0))$ for all $a_i \in V_8$, $i = 0, 1, \ldots, 15$, with addition and multiplication in the field P.

2.2.3 Transformations

Transformations involved in encryption/decryption processes are described as follows:

$$\begin{array}{ll}
X[k] : V_{128} \to V_{128}, & X[k](a) = k \oplus a,\ k, a \in V_{128}; \\
S : V_{128} \to V_{128}, & S(a) = S(a_{15} \| \ldots \| a_0) = \pi(a_{15}) \| \ldots \| \pi(a_0), \\
 & \text{where } a = a_{15} \| \ldots \| a_0 \in V_{128},\ a_i \in V_8,\ i = 0, 1, \ldots, 15;
\end{array}$$

$S^{-1} : V_{128} \to V_{128}$, an inverse transformation for S, which could be evaluated, for example, as follows: $S^{-1}(a) = S^{-1}(a_{15}\| \dots \|a_0) = \pi^{-1}(a_{15})\| \dots \|\pi^{-1}(a_0)$, where $a = a_{15}\| \dots \|a_0 \in V_{128}$, $a_i \in V_8$, $i = 0, 1, \dots, 15$, π^{-1} – an inverse substitution for π;

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

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

$R^{-1} : V_{128} \to V_{128}$, an inverse transformation for R, which could be evaluated, for example, as follows: $R^{-1}(a) = R^{-1}(a_{15}\| \dots \|a_0) =$
$= a_{14}||a_{13}|| \dots ||a_0||l(a_{14}, a_{13}, \dots, a_0, a_{15})$,
where $a = a_{15}\| \dots \|a_0 \in V_{128}$, $a_i \in V_8$, $i = 0, 1, \dots, 15$;

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

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

2.3 Key schedule

The round constants $C_i \in V_{128}$, $i = 1, 2, \dots, 32$, defined as

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

are used in the key schedule.

The round keys $K_i \in V_{128}$, $i = 1, 2, \dots, 10$, are derived from a master-key $K \in V_{256}$ as follows:

$$K_1\|K_2 = K;$$

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

2.4 Encryption algorithm

A 128-bit plain text $a \in V_{128}$ is encrypted as follows:

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

2.5 Decryption algorithm

A 128-bit cipher text $b \in V_{128}$ is decrypted as follows

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

References

[1] GOST 28147-89. Information Processing Systems. Cryptographic Protection. Cryptographic Transformation Algorithm (In Russian).

[2] A. Poschmann, S. Ling, H. Wang. 256 Bit Standardized Crypto for 650 GE - GOST Revisited // CHES 2010. — LNCS 6225. — P. 219–233.

[3] T. Isobe. A Single-Key Attack on the Full GOST Block Cipher // FSE 2011. — LNCS 6733. — P. 290–305.

[4] I. Dinur, O. Dunkelman, A. Shamir. Improved Attacks on Full GOST. Cryptology ePrint Archive, Report 2011/558, 2011. http://eprint.iacr.org/2011/558.

[5] ISO/IEC JTC 1/SC 27 Standing Document 12 (SD12) on the Assessment of cryptographic algorithms and key lengths. — http://www.jtc1sc27.din.de/sbe/SD12

[6] D. Dygin, A. Dmukh, G. Marshalko. A lightweight-friendly modification of GOST block cipher // Pre-proc. CTCrypt 2013. — P. 51–61. — To appear in Mathematical Aspects of Cryptography.

[7] V.A. Shishkin. Design principles of a prospective block cipher with 128 bit block length. — Presentation at RusCrypto'2013 (In Russian). — http://www.ruscrypto.ru/resource/summary/rc2013.