Independent Submission
Request for Comments: 7801
Category: Informational
ISSN: 2070-1721

V. Dolmatov, Ed.
Research Computer Center MSU

March 2016

ГОСТ Р 34.12-2015 — Блочный шифр «Кузнечик»

GOST R 34.12-2015: Block Cipher "Kuznyechik"

Тезисы

Этот документ публикуется в качестве источника информации о Национальном стандарте Российской федерации ГОСТ Р 34.12-2015, описывающем блочное шифрование с размером блока n=128 битов и размером ключа k=256 битов, называемого также «Кузнечик». Данный алгоритм является одним из множества российских стандартных криптографических алгоритмов (алгоритмы ГОСТ).

Статус документа

Документ не является спецификацией стандарта Internet и публикуется с информационными целями.

Документ включается в серию RFC, но не связан с каким-либо потоком документов RFC. Редактор (RFC Editor) принял решение о публикации документа по своему усмотрению и не делает каких-либо заявления о реализации или развертывании. Документы, одобренные для публикации их редакторами не претендуют на какой-либо уровень стандартизации Internet (см. раздел 2 в RFC 5741).

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке http://www.rfc-editor.org/info/rfc7801.

Оглавление

1. Сфера действия стандарта

Национальный стандарт Российской федерации [GOST3412-2015] описывает базовые блочные шифры, используемые в качестве криптографических методов обработки и защиты информации, включая защиту конфиденциальности, достоверности и целостности информации при ее передаче, обработке и хранении в компьютерных системах.

Криптографические алгоритмы, описанные в этом стандарте, предназначены как для программно, так и для аппаратной реализации. Они соответствуют современным криптографическим требованиям и не накладывают каких-либо ограничений на уровень конфиденциальности защищаемой информации.

Стандарт применяется при разработке, использовании и модернизации информационных систем разного назначения.

2. Общие сведения

Блочный шифр «Кузнечик» [GOST3412-2015] Разработан Центром защиты информации и спефиальной связи Федеральной службы безопасности Российской федерации с участием ОАО «Инфотекс» («Информационные Технологии и Коммуникационные Системы» - InfoTeCS JSC). Стандарт ГОСТ Р 34.12-2015 одобрен и введен в действие Приказом №749-ст Федерального агентства по техническому регулированию и метрологии от 19 июня 2015 г.

Терминологически и концептуально стандарт согласован с международными стандартами:

- ISO/IEC 10116 [ISO-IEC10116];
- серия стандартов ISO/IEC 18033 [ISO-IEC18033-1] [ISO-IEC18033-3].

3. Определения и обозначения

Ниже приведены используемые в стандарте термины и их определения.

3.1. Определения

Ниже приведены определения используемых в стандарте терминов.

encryption algorithm — алгоритм шифрования
Процесс, преобразующий открытый текст в зашифрованный (параграф 2.19 [ISO-IEC18033-1]).

decryption algorithm — алгоритм дешифрования
Процесс, преобразующий зашифрованный текст в открытый (параграф 2.14 [ISO-IEC18033-1]).

basic block cipher — базовый блочный шифр
Блочный шифр, в котором для данного ключа выполняется однократное обратимое отображение набора блоков открытого текста с фиксированным размером в блоки шифрованного текста с таким же размером.

block - блок
Строка битов определенного размера (параграф 2.6 [ISO-IEC18033-1])

block cipher – блочный шифр
Симметричная система шифрования, в которой алгоритм шифрования применяется к блокам открытого текста (строкам битов определенного размера) для получения блоков шифрованного текста (параграф 2.7 [ISO-IEC18033-1]).
Примечание. В ГОСТ Р 34.12-2015 установлено, что термины «блочный шифр» (block cipher) и «алгоритм блочного шифрования» (block encryption algorithm) являются синонимами.

encryption - шифрование
Обратимое преобразование данных с помощью криптографического алгоритма для создания шифрованного текста, т. е., сокрытия содержимого этих данных (параграф 2.18 [ISO-IEC18033-1]).

round key — итерационный ключ
Последовательность символов, рассчитываемая в процессе развертывания ключа и определяющая преобразование в рамках одной итерации (round) блочного шифрования.

key - ключ
Последовательность символов, определяющая криптографическое преобразование (например, шифрование или дешифрование) (параграф 2.21 [ISO-IEC18033-1]).
Примечание. В ГОСТ Р 34.12-2015 ключ должен быть двоичной последовательностью.

plaintext - открытый текст
Незашифрованная информация (параграф 3.11 [ISO-IEC10116]).

key schedule - развертывание ключа
Расчет итерационного ключа на основе ключа шифра.

decryption - дешифрование
Операция, обратная шифрованию (параграф 2.13 [ISO-IEC18033-1]).

symmetric cryptographic technique — симметричный криптографический метод
Криптографический метод, в котором используется один и тот же секретный ключ как для шифрования, так и для дешифрования (параграф 2.32 [ISO-IEC18033-1]).

cipher - шифр
Другой термин, используемый для обозначения криптографической системы (параграф 2.20 [ISO-IEC18033-1]).

ciphertext — шифрованный текст
Данные, которые были преобразованы с целью сокрытия их содержимого (параграф 3.3 [ISO-IEC10116]).

3.2. Обозначения

Ниже приводятся используемые в стандарте обозначения.

V*
множество всех двоичных строк конечного размера (далее, строк), включая пустую строку.

V_s
множество всех двоичных строк размера s, где s - неотрицательное целое число; нумерация подстрок и компонент строки осуществляется справа налево, начиная с 0.

U[*]W
прямое (декартово) произведение множеств U и W.

|A|
число компонент (размер) строки A из множества V* (если A - пустая строка, |A| = 0).

A||B
конкатенация строка A и B из множества V*, т.е., строка из V_(|A|+|B|), где левая подстрока из V_|A| равна A, а правая подстрока из V_|B| равна B.

Z_(2^n)
кольцо вычетов по модулю 2^n,

Q
конечное поле GF(2)[x]/p(x), где $p(x)=x^8+x^7+x^6+x+1$ принадлежит GF(2)[x]; элементы поля Q представлены целыми числами так, что элемент $z\_0+z\_1*theta+...+z\_7*theta^7$, принадлежащий Q, соответствует целому числу $z\_0+2*z\_1+...+2^7*z\_7$, где z_i=0 или z_i=1, i=0,1,...,7, а theta означает класс вычетов по модулю p(x), содержащий x.

(xor)
исключающее ИЛИ для двух двоичных строк одинаковых размеров.

Vec_s: Z_(2^s) -> V_s
взаимно-однозначное[1] отображение, сопоставляющее элемент из кольца ring $Z_{(2^s)}$ с его двоичным представлением; т.е., для любого элемента z в кольце $Z_{(2^s)}$, представленного в виде $z\_0 + (2*z\_1) + ... + (2^{(s-1)}*z\_(s-1))$, где z_i принадлежит {0, 1}, i = 0, ..., n-1, справедливо равенство $Vec\_s(z) = z\_(s-1)||...||z\_1||z\_0$.

Int_s: V_s -> Z_(2^s)
отображение, обратное по отношению к Vec_s (т.е., $Int\_s = Vec\_s^{-1}$).

delta: V_8 -> Q
взаимно-однозначное отображение, сопоставляющее двоичную строку из V_8 с элементом из поля Q следующим образом — строке $z\_7||...||z\_1||z\_0$, где z_i принадлежит {0, 1}, i = 0, ..., 7, соответствует элемент $z\_0+(z\_1*theta)+...+(z\_7*theta^7)$, принадлежащий Q[2],

nabla: Q -> V8
отображение, обратно по отношению к delta, т.е., $delta = nabla^{-1}$,

PS
композиция отображений, в которой отображение S выполняется первым.

P^s
композиция отображений P^{s-1} и P, где $P^1=P$.

4. Значения параметров

4.1. Нелинейное взаимно-однозначное преобразование

Нелинейное взаимно-однозначное преобразование представляет собой подстановку Pi = (Vec_8)Pi'(Int_8): V_8 -> V_8, где Pi': $Z_{(2^8)}$ -> $Z_{(2^8)}$. Значения для подстановки Pi' приведены ниже в форме массива Pi' = (Pi'(0), Pi'(1), ... , 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)
```

Pi^{-1} представляет собой значение, обратное Pi (1/Pi); значения для подстановки Pi^{-1}' приведены ниже в форме массива Pi^{-1}' = (Pi^{-1}'(0), Pi^{-1}'(1), ... , Pi^{-1}'(255)):

```
 Pi^-1' =
(     165,  45,  50, 143,  14,  48,  56, 192,  84, 230, 158,
       57,  85, 126,  82, 145, 100,   3,  87,  90,  28,  96,
        7,  24,  33, 114, 168, 209,  41, 198, 164,  63, 224,
       39, 141,  12, 130, 234, 174, 180, 154,  99,  73, 229,
       66, 228,  21, 183, 200,   6, 112, 157,  65, 117,  25,
      201, 170, 252,  77, 191,  42, 115, 132, 213, 195, 175,
       43, 134, 167, 177, 178,  91,  70, 211, 159, 253, 212,
       15, 156,  47, 155,  67, 239, 217, 121, 182,  83, 127,
      193, 240,  35, 231,  37,  94, 181,  30, 162, 223, 166,
      254, 172,  34, 249, 226,  74, 188,  53, 202, 238, 120,
        5, 107,  81, 225,  89, 163, 242, 113,  86,  17, 106,
      137, 148, 101, 140, 187, 119,  60, 123,  40, 171, 210,
       49, 222, 196,  95, 204, 207, 118,  44, 184, 216,  46,
       54, 219, 105, 179,  20, 149, 190,  98, 161,  59,  22,
      102, 233,  92, 108, 109, 173,  55,  97,  75, 185, 227,
```

[1]Используется также термин «биективное». *Прим. перев.*
[2]В исходном документе ошибочно сказано «Z». См. RFC Errata. *Прим. перев.*

```
        186, 241, 160, 133, 131, 218,  71, 197, 176,  51, 250,
        150, 111, 110, 194, 246,  80, 255,  93, 169, 142,  23,
         27, 151, 125, 236,  88, 247,  31, 251, 124,   9,  13,
        122, 103,  69, 135, 220, 232,  79,  29,  78,   4, 235,
        248, 243,  62,  61, 189, 138, 136, 221, 205,  11,  19,
        152,   2, 147, 128, 144, 208,  36,  52, 203, 237, 244,
        206, 153,  16,  68,  64, 146,  58,   1,  38,  18,  26,
         72, 104, 245, 129, 139, 199, 214,  32,  10,   8,   0,
         76, 215, 116 )
```

4.2. Линейное преобразование

Линейное преобразование обозначается l: $(V\_8)^{16}$ -> V_8 и определеяется выражением

```
l(a_15,...,a_0) = nabla(148*delta(a_15) + 32*delta(a_15) + 133*delta(a_13) + 16*delta(a_12) +
194*delta(a_11) + 192*delta(a_10) + 1*delta(a_9) + 251*delta(a_8) + 1*delta(a_7) +
192*delta(a_6) + 194*delta(a_5) + 16*delta(a_4) + 133*delta(a_3) + 32*delta(a_2) +
148*delta(a_1) +1*delta(a_0))
```

для всех a_i, принадлежащих V_8, i = 0, 1, ..., 15, где операции сложения и умножения выполняются в поле Q, а постоянные являются элементами поля, как указано выше.

4.3. Преобразования

Для реализации алгоритмов шифрования и дешифрования используются следующие преобразования:

```
X[x]:V_128->V_128  X[k](a)=k(xor)a
```

где k принадлежит V_128,

```
S:V_128-> V_128  S(a)=(a_15||...||a_0)=pi(a_15)||...||pi(a_0)
```

где a_15||...||a_0 принадлежит V_128, a_i принадлежит V_8, i=0,1,...,15,

S^{-1}:V_128-> V_128

значение, обратное по отношению к результату преобразования S, которое можно выполнить, как показано ниже

$S^{-1}(a\_15||...||a\_0)=pi^{-1}(a\_15)||...||pi^{-1}(a\_0)$

где a_15||...||a_0 принадлежит V_128, a_i принадлежит V_8, i=0,1,...,15,

```
R:V_128-> V_128  R(a_15||...||a_0)=l(a_15,...,a_0)||a_15||...||a_1
```

где a_15||...||a_0 принадлежит V_128, a_i принадлежит V_8, i=0,1,...,15,

$L:V\_128\text{-}>V\_128\ \ L(a)=R^{16}(a)$

где a принадлежит V_128,

R^{-1}:V_128-> V_128

значение, обратное по отношению к результату преобразования R, которое можно выполнить, как показано ниже

$R^{-1}(a\_15||...||a\_0)=a\_14||\ a\_13||...||a\_0||l(a\_14,a\_13,...,a\_0,a\_15)$

где a_15||...||a_0 принадлежит V_128, a_i принадлежит V_8, i=0,1,...,15,

$L^{-1}:V\_128\text{-}>V\_128\ \ L^{-1}(a)=(R^{-1})(16)(a)$

где a принадлежит V_128,

```
F[k]:V_128[*]V_128 -> V_128[*]V_128 F[k](a_1,a_0)=(LSX[k](a_1)(xor)a_0,a_1)
```

где k, a_0, a_1 принадлежат V_128.

4.4. Развертывание ключа

При развертывании ключа используются итерационные константы C_i принадлежащие V_128, i=1, 2, ..., 32, и определяемые, как

```
C_i=L(Vec_128(i))
    i=1,2,...,32
```

Итерационные ключи Round keys _i, i=1, 2, ..., 10 создаются на основе ключа K=k_255||...||k_0 принадлежащего V_256, k_i принадлежащих V_1, i=0, 1, ..., 255, как показано ниже

```
K_1=k_255||...||k_128
K_2=k_127||...||k_0
(K_(2i+1),K_(2i+2))=F[C_(8(i-1)+8)]...F[C_(8(i-1)+1)](K_(2i-1),K_(2i))
    i=1,2,3,4
```

4.5. Базовый алгоритм шифрования

4.5.1. Шифрование

В зависимости от значений итерационных ключей K_1,...,K_10 алгоритм шифрования определяется подстановкой E_(K_1,...,K_10), определяемой как:

```
E_(K_1,...,K_10)(a)=X[K_10]LSX[K_9]...LSX[K_2]LSX[K_1](a)
```

где a принадлежит V_128.

4.5.2. Дешифрование

В зависимости от значений итерационных ключей K_1,...,K_10 алгоритм дешифрования определяется подстановкой D_(K_1,...,K_10), определяемой как:

$D\_(K\_1,...,K\_10)(a)=X[K\_1]L^{-1}S^{-1}X[K\_2]...\ L^{-1}S^{-1}X[K\_9]\ L^{-1}S^{-1}X[K\_10](a)$

где a принадлежит V_128.

5. Примеры (для информации)

Приведенная в этом разделе информация не относится к нормативной части стандарта.

5.1. Преобразование S

```
S(ffeeddccbbaa99881122334455667700) = b66cd8887d38e8d77765aeea0c9a7efc
S(b66cd8887d38e8d77765aeea0c9a7efc) = 559d8dd7bd06cbfe7e7b262523280d39
S(559d8dd7bd06cbfe7e7b262523280d39) = 0c3322fed531e4630d80ef5c5a81c50b
S(0c3322fed531e4630d80ef5c5a81c50b) = 23ae65633f842d29c5df529c13f5acda
```

5.2. Преобразование R

```
R(00000000000000000000000000000100) = 94000000000000000000000000000001
R(94000000000000000000000000000001) = a5940000000000000000000000000000
R(a5940000000000000000000000000000) = 64a59400000000000000000000000000
R(64a59400000000000000000000000000) = 0d64a594000000000000000000000000
```

5.3. Преобразование L

```
L(64a59400000000000000000000000000) = d456584dd0e3e84cc3166e4b7fa2890d
L(d456584dd0e3e84cc3166e4b7fa2890d) = 79d26221b87b584cd42fbc4ffea5de9a
L(79d26221b87b584cd42fbc4ffea5de9a) = 0e93691a0cfc60408b7b68f66b513c13
L(0e93691a0cfc60408b7b68f66b513c13) = e6a8094fee0aa204fd97bcb0b44b8580
```

5.4. Развертывание ключа

В этом примере используется ключ

```
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]...F[C_1](K_1, K_2) = (f9eae5f29b2815e31f11ac5d9c29fb01,
       37777748e56453377d5e262d90903f87)

C_4 = 7bcd1b0b73e32ba5b79cb140f2551504
F[C_4]...F[C_1](K_1, K_2) = (e980089683d00d4be37dd3434699b98f,
       f9eae5f29b2815e31f11ac5d9c29fb01)

C_5 = 156f6d791fab511deabb0c502fd18105
F[C_5]...F[C_1](K_1, K_2) = (b7bd70acea4460714f4ebe13835cf004,
       e980089683d00d4be37dd3434699b98f)

C_6 = a74af7efab73df160dd208608b9efe06
F[C_6]...F[C_1](K_1, K_2) = (1a46ea1cf6ccd236467287df93fdf974,
       b7bd70acea4460714f4ebe13835cf004)

C_7 = c9e8819dc73ba5ae50f5b570561a6a07
F[C_7]...F [C_1](K_1, K_2) = (3d4553d8e9cfec6815ebadc40a9ffd04,
       1a46ea1cf6ccd236467287df93fdf974)

C_8 = f6593616e6055689adfba18027aa2a08
(K_3, K_4) = F [C_8]...F [C_1](K_1, K_2) = (db31485315694343228d6aef8cc78c44,
        3d4553d8e9cfec6815ebadc40a9ffd04)
```

Итерационные ключи K_i для i = 1, 2, ..., 10 имеют значения:

```
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
```

5.5. Тестовое шифрование

В этом примере выполняется тестовое шифрование с использованием итерационных ключей, указанных в параграфе 5.4. Пусть открытый текст имеет вид

```
a = 1122334455667700ffeeddccbbaa9988
```

тогда

```
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
```

Зашифрованный текст будет иметь вид

```
b = X[K_10]LSX[K_9]...LSX[K_1](a) = 7f679d90bebc24305a468d42b9d4edcd
```

5.6. Тестовая расшифровка

В этом примере выполняется тестовое дешифрование с использованием итерационных ключей, указанных в параграфе 5.4. Пусть шифрованный текст имеет вид

```
b = 7f679d90bebc24305a468d42b9d4edcd
```

тогда

```
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
```

Расшифрованный текст имеет вид

```
a = X[K_1]S^(-1)L^(-1)X[K_2]...S^(-1)L^(-1)X[K_10](b) = 1122334455667700ffeeddccbbaa9988
```

6. Вопросы безопасности

Документ целиком посвящен вопросам безопасности.

7. Литература

7.1. Нормативные документы

[GOST3412-2015] "Information technology. Cryptographic data security. Block ciphers", GOST R 34.12-2015, Federal Agency on Technical Regulating and Metrology, 2015.

7.2. Дополнительная литература

[ISO-IEC10116] ISO/IEC, "Information technology -- Security techniques – Modes of operation for an n-bit block cipher", ISO/IEC 10116, 2006.

[ISO-IEC18033-1] ISO/IEC, "Information technology -- Security techniques – Encryption algorithms -- Part 1: General", ISO/IEC 18033-1, 2015.

[ISO-IEC18033-3] ISO/IEC, "Information technology -- Security techniques – Encryption algorithms -- Part 3: Block ciphers", ISO/IEC 18033-3, 2010.

Адрес автора

Василий Долматов (редактор)

Research Computer Center MSU

Leninskiye Gory, 1, Building 4, MGU NIVC

Moscow 119991

Russian Federation

Email: dol@srcc.msu.ru

Перевод на русский язык

Николай Малых

nmalykh@gmail.com