Independent Submission
Request for Comments: 5830
Category: Informational
ISSN: 2070-1721

V. Dolmatov, Ed.
Cryptocom, Ltd.

March 2010

ГОСТ 28147-89. Алгоритмы шифрования, дешифрования и MAC

GOST 28147-89: Encryption, Decryption, and Message Authentication Code (MAC) Algorithms

Аннотация

Этот документ опубликован в качестве источника информации о стандарте Российской Федерации для алгоритмов шифрования, дешифрования и идентификации сообщений (ГОСТ 28147-89), который является одним из российских стандартов для криптографических алгоритмов, называемых алгоритмами ГОСТ. Недавно российские криптоалгоритмы начали использовать в приложениях Internet и этот документ был подготовлен в качестве источника информации для разработчиков и пользователей алгоритмов ГОСТ 28147-89 в плане шифрования, дешифровки и идентификации сообщений.

Статус документа

Этот документ не является спецификацией проекта стандарта Internet и публикуется с информационными целями.

Этот документ подготовлен независимо от остальных документов серии RFC. Редактор документа (RFC Editor) был выбран для публикации документа по его собственному усмотрению и не делает каких-либо заявлений о значимости документа для реализации или развёртывания. Документ одобрен для публикации редактором и не претендует на роль какого-либо стандарта Internet (см. параграф 2 документа RFC 5741).

Информацию о текущем состоянии документа, обнаруженных ошибках и способах связи с разработчиками можно найти по ссылке http://www.rfc-editor.org/info/rfc5830.

Оглавление

1. Введение....1
1.1. Общие сведения....1
2. Применимость....2
3. Определения и обозначения....2
3.1. Определения....2
3.2. Обозначения....2
4. Общие сведения....2
5. Режим простой замены....3
5.1. Зашифровка открытых данных в режиме простой замены....3
5.2. Дешифровка в режиме простой замены....4
6. Режим гаммирования....5
6.1. Зашифровка открытых данных в режиме гаммирования....5
6.2. Расшифровка данных с режиме гаммирования....6
7. Режим гаммирования с обратной связью....6
7.1. Зашифровка данных в режиме гаммирования с обратной связью....6
7.2. Расшифровка в режиме гаммирования с обратной связью....7
8. Режим генерации MAC....7
9. Вопросы безопасности....8
10. Нормативные документы....8
Приложение A. Значения констант C1 и C2....8
Приложение B. Разработчики документа....9

1. Введение

1.1. Общие сведения

[GOST28147-89] является алгоритмом унифицированного криптографического преобразования для систем обработки информации разного назначения, определяющим правила шифрования/дешифрования и генерации кода идентификации сообщений (MAC[1]).

[1]Message authentication code.

Алгоритм криптографических преобразований предназначен для аппаратной или программной реализации и соответствует криптографическим требованиям. Алгоритм не вносит ограничений на уровень секретности шифруемой информации.

2. Применимость

ГОСТ 28147-89 определяет модель шифрования/дешифрования и генерации MAC для заданного сообщения (документа), который предназначен для передачи по незащищённым публичным телекоммуникационным каналам между системами обработки данных разного назначения.

Алгоритм ГОСТ 28147-89 обязателен в Российской Федерации для использования во всех системах обработки данных, предоставляющих публичный сервис.

3. Определения и обозначения

3.1. Определения

Ниже приведены определения используемых в стандарте терминов.

Running key - рабочий ключ[1]
Псевдослучайная последовательность битов, генерируемая по заданному алгоритму, для шифрования открытых данных и дешифровки шифрованных данных.

Encryption - шифрование[2]
Процесс преобразования открытых данных в зашифрованные с помощью шифра.

MAC - код идентификации сообщения[3]
Блок информации фиксированного размера, который создаётся по некому правилу из открытых данных и ключа и добавляется к зашифрованным данным для защиты от фальсификации данных.

Key - ключ
Определённое секретное состояние некоторых параметров алгоритма криптографического преобразования, которое обеспечивает выбор одного преобразования из совокупности всех возможных.

Cryptographic protection - криптографическая защита, криптозащита
Защита данных с помощью их криптографического преобразования.

Cryptographic transformation – криптографическое преобразование
Преобразование данных с использованием шифрования и (или) MAC.

Decryption - дешифровка[4]
Процесс преобразования защищённых данных в открытые с использованием шифра.

Initialisation vector - вектор инициализации[5]
Исходные значения открытых параметров алгоритма криптографического преобразования.

Encryption equation - уравнение зашифровки
Соотношение, выражающее процесс получения зашифрованных данных из открытых в результате преобразований, заданных алгоритмом криптографического преобразования.

Decryption equation - уравнение расшифровки
Соотношение, выражающее процесс получения открытых данных из зашифрованных в результате преобразований, заданных алгоритмом криптографического преобразования.

Cipher - шифр
Набор обратимых преобразований множества возможных открытых данных во множество возможных зашифрованных данных, выполняемых по заданным правилам с применением ключей[6].

3.2. Обозначения

В этом документе используются следующие обозначения:

(+) побитовое сложение слов одного размера по модулю 2;

[+] сложение 32-битовых векторов по модулю 2^{32}.

[+]' сложение 32-битовых векторов по модулю 2^{32}-1.

1..N множество целых чисел от 1 до N, включительно.

4. Общие сведения

Структурная схема алгоритма криптографических преобразований (криптографическая модель[7]) включает:

- 256-битовое устройство хранения ключей (KDS9), состоящее из восьми 32-битовых регистров (X0, X1, X2, X3, X4, X5, X6, X7);
- четыре 32-битовых регистра (N1, N2, N3, N4);
- два 32-битовых регистра (N5 и N6), содержащих константы[8] C1 и C2;
- два 32-битовых сумматора по модулю 2^{32} (CM1, CM3);
- 32-битовый сумматор для побитового сложения по модулю 2 (CM2);
- 32-битовый сумматор по модулю (2^{32}-1) (CM4);

[1] В исходном стандарте используется термин «гамма шифра». *Прим. перев.*
[2] В исходном стандарте используется термин «зашифрование». *Прим. перев.*
[3] В исходном стандарте используется термин «имитовставка». *Прим. перев.*
[4] В исходном стандарте используется термин «расшифрование данных». *Прим. перев.*
[5] В исходном стандарте используется термин «синхропосылка». *Прим. перев.*
[6] В оригинале это предложение содержит ошибку. См. https://www.rfc-editor.org/errata_search.php?eid=2137. *Прим. перев.*
[7] В исходном стандарте используется термин «криптосхема». *Прим. перев.*
[8] В исходном стандарте используется термин «постоянные запоминания». *Прим. перев.*

- сумматор по модулю 2 (CM5) без ограничения по разрядности;
- блок подстановки (K);
- циклический регистр сдвига на одиннадцать разрядов[1] в сторону старшего разряда (R).

Блок подстановок (S-box) K состоит из восьми узлов замены K1, K2, K3, K4, K5, K6, K7, K8 с памятью по 64 бита в каждом. Приходящий в блок подстановки 32-битовый вектор делится на 8 последовательных 4-битовых векторов, каждый из которых преобразуется соответствующим узлом замены в другой 4-битовый вектор. Узел замены представляет собой таблицу из 16 строк по 4 бита. Входной вектор определяет номер строки в таблице, а содержимое этой строки служит выходным вектором. Далее 4-битовые выходные векторы последовательно объединяются в 32-битовый вектор.

Примечание. Стандарт не определяет каких-либо блоков подстановки. Некоторые из таких блоков определены в [RFC4357].

При сложении и циклическом сдвиге двоичных векторов старшими считаются разряды с большими номерами.

При записи ключа (W1, W2, ..., W256), Wq = 0..1, q = 1..256 в KDS:

- значение W1 записывается в первый бит регистра X0;
- значение W2 записывается во второй бит регистра X0;
- ...
- значение W32 записывается в 32-й бит регистра X0;
- значение W33 записывается в первый бит регистра X1;
- значение W34 записывается во второй бит регистра X1;
- ...
- значение W64 записывается в 32-й бит регистра X1;
- значение W65 записывается в первый бит регистр X2;
- ...
- значение W256 записывается в 32-й бит регистра X7.

При перезаписи информации значение p-го бита одного регистра (сумматора) записывается в p-й бит другого регистра (сумматора).

Значения констант C1 и C2, хранящихся в регистрах N5 и N6 приведены в приложении 1.

Ключи, определяющие заполнение KDS и таблиц блока подстановки K, являются секретными элементами и поставляются в установленном порядке.

Заполнение блока подстановки K описано в ГОСТ 28147-89, как долговременный ключевой элемент, который является общим для компьютерной сети. Обычно K используется в качестве параметра алгоритма и некоторые возможные наборы значений K описаны в [RFC4357].

В криптографической модели предполагается 4 режима работы:

- зашифровка (дешифровка) данных в режиме простой замены (ECB[2]);
- зашифровка (дешифровка) данных в режиме гаммирования (CNT[3]);
- зашифровка (дешифровка) данных в режиме гаммирования с обратной связью (CFB5);
- генерация MAC (имитовставки).

В [RFC4357] также описан режим CBC, но он не входит в стандарт. =

5. Режим простой замены

5.1. Зашифровка открытых данных в режиме простой замены

Шифруемые данные делятся на блоки по 64 бита в каждом. Ввод любого двоичного блока Tp = (a1(0), a2(0), ... , a31(0), a32(0), b1(0), b2(0), ..., b32(0)) в регистры N1 и N2 выполняется таким образом, что значение a1(0) помещается в первый разряд регистра N1, значение a2(0) - во второй разряд N1 и т. д, а значение a32(0) помещается в 32-й разряд регистра N1. Значение b1(0) помещается в первый разряд регистра N2, значение b2(0) - во второй разряд N2, ..., значение b32(0) - в 32-й разряд регистра N2[4].

В результате получается состояние (a32(0), a31(0), ..., a2(0), a1(0)) в регистре N1 и состояние (b32(0), b31(0), ..., b1(0)) в регистре N2.

В KDS вводятся 256 битов ключа. Содержимое восьми 32-битовых регистров X0, X1, ..., X7 будет иметь вид:

```
X0 = W32, W31, ..., W2, W1
X1 = W64, W63, ..., W34, W33
. . . . . .
```

[1] В исходном стандарте не вполне корректно сказано «шагов», а не «разрядов». *Прим. перев.*
[2] Electronic codebook.
[3] Counter mode.
[4] В оригинале этот абзац содержит ошибку. См. https://www.rfc-editor.org/errata_search.php?eid=2139. *Прим. перев.*

```
X7 = W256, W255, ..., W226, W225
```

Алгоритм зашифровки 64-битового блока открытых данных в режиме простой замены состоит из 32 циклов.

В первом цикле начальное значение регистра N1 складывается по модулю 2^{32} в сумматоре СМ1 с содержимым регистра X0 блока хранения ключей. Отметим, что значение регистра N1 при этом остаётся неизменным.

Результат суммирования преобразуется в блоке подстановки K и полученный вектор помещается в регистр R, где он циклически сдвигает на 11 разрядов в направлении старших битов[1]. Результат операции сдвига суммируется поразрядно по модуля 2 в сумматоре СМ2 с 32-битовым значением регистра N2. Полученные в сумматоре СМ2 результат записывается в регистр N1, а прежнее содержимое этого регистра переносится в регистр N2. На этом первый цикл заканчивается.

Последующие циклы выполняются по аналогии с первым. При этом:

- во втором цикле из KDS считывается[2] содержимое регистра X1;
- в третьем цикле из KDS считывается содержимое регистра X2;
- ...
- в восьмом цикле из KDS считывается содержимое регистра X7.
- в циклах 9 - 16 и 17 - 24 операции считывания содержимого регистров KDS повторяются в том же порядке:

```
X0, X1, X2, X3, X4, X5, X6, X7.
```

- в последних 8 циклах (25 - 32) содержимое регистров KDS считывается в обратном порядке:

```
X7, X6, X5, X4, X3, X2, X1, X0.
```

Таким образом, в 32 циклах зашифровки выполняется следующий порядок выборки содержимого регистров KDS:

```
X0, X1, X2, X3, X4, X5, X6, X7, X0, X1, X2, X3, X4, X5, X6, X7,
X0, X1, X2, X3, X4, X5, X6, X7, X7, X6, X5, X4, X3, X2, X1, X0
```

В 32-м цикле результат из сумматора СМ2 копируется в регистр N2, а в регистре N1 значение сохраняется.

После выполнения 32-го цикла в регистрах N1 и N2 будет содержаться зашифрованный блок данных, соответствующий блоку открытых данных.

Уравнения для зашифровки в режиме electronic codebook имеют вид:

```
|a(j) = (a(j-1) [+] X(j-1)(mod 8))*K*R (+) b (j-1)
|b(j) = a(j-1)
```

при j = 1..24;

```
|a(j) = (a(j-1) [+] X(32-j))*K*R (+) b(j-1)
|b(j) = a(j-1)
```

при j = 25..31; a32 = a31;

```
b(32) = (a(31) [+] X0)*K*R (+) b(31)
```

при j=32.

Где:

a(0) = (a32(0), a31(0), ..., a1(0)) - начальные значения регистра N1 перед первым циклом зашифровки;

b(0) = (b32(0), b31(0), ..., b1(0)) - начальные значения регистра N2 перед первым циклом зашифровки;

a(j) = (a32(j), a31(j), ..., a1(j)) - содержимое регистра N1 после j-го цикла зашифровки (j = 1..32);

b(j) = (b32(j), b31(j), ..., b1(j)) - содержимое регистра N2 после j-го цикла зашифровки (j = 1..32).

R - операция циклического сдвига на 11 разрядов в направлении старших битов:

```
R(r32, r31, r30, r29, r28, r27, r26, r25, r24, r23, r22, r21, r20, ..., r2, r1) =
(r21, r20, ..., r2, r1, r32, r31, r30, r29, r28, r27, r26, r25, r24, r23, r22)
```

64-битовый блок зашифрованных данных Tc считывается из регистров N1 и N2 в следующем порядке:

1-й, 2-й, ..., 32-й бит регистра N1, затем 1-й, 2-й, ..., 32-й бит регистра N2

```
Tc = a1(32), a2(32), ..., a32(32), b1(32), b2(32), ..., b32(32)).
```

Остальные блоки открытых данных зашифровываются в режиме простой замены аналогично описанному.

5.2. Дешифровка в режиме простой замены

Ключ, который использовался при зашифровке (256 битов) загружается в KDS, зашифрованные данные делятся на блоки по 64 бита. Ввод любого двоичного блока

```
Tc = (a1(32), a2(32), ..., a32(32), b1(32), b2(32), ..., b32(32))
```

в регистры N1 и N2 выполняется следующим образом:

- значение a1(32) записывается в первый бит регистра N1;
- значение a2(32) записывается во второй бит регистра N1;
- ...
- значение a32(32) записывается в 32-й бит регистра N1;
- значение b1(32) записывается в первый бит регистра N2;

[1] Влево в привычной терминологии. *Прим. перев.*
[2] Для суммирования. *Прим. перев.*

- ...
- значение b32(32) записывается в 32-й бит регистра N2.

Процедура дешифровки использует тот же алгоритм, который применялся при зашифровке открытых данных с одним исключением - содержимое регистров X0, X1, ..., X7 считывается из KDS для дешифровки в следующем порядке:

```
X0,X1,X2,X3,X4,X5,X6,X7, X7,X6,X5,X4,X3,X2,X1,X0,
X7,X6,X5,X4,X3,X2,X1,X0, X7,X6,X5,X4,X3,X2,X1,X0.
```

Уравнение дешифровки имеет вид:

```
|a(32-j) = (a(32-j+1) [+] X(j-1))*K*R (+) b(32-j+1)
|b(32-1) = a(32-j+1)
```

при j = 1..8;

```
|a(32-j) = (a(32-j+1) [+] X(j-1)(mod 8))*K*R (+) b(32-j+1)
|b(32-1) = a(32-j+1)
```

при j = 9..31;

```
|a(0) = a(1)
|b(0) = (a(1) [+] X0)*K*R (+) b1
```

при j=32.

Содержимое регистров N1 и N2 после 32 циклов работы составляет блок расшифрованных (открытых) данных[1].

```
Tp = (a1(0), a2(0), ... , a32(0), b1(0), b2(0), ..., b32(0))
```

соответствующий зашифрованному блоку:

- значение a1(0) блока Tp соответствует первому биту регистра N1;
- значение a2(0) соответствует второму биту регистра N1;
- ...
- значение b1(0) соответствует первому биту регистра N2;
- значение b2(0) соответствует второму биту регистра N2;
- ...
- значение b32(0) соответствует 32-му биту регистра N2.

Оставшиеся блоки расшифровываются аналогично.

Алгоритм зашифровки в режиме простой замены 64-битового блока Tp обозначается A, т. е.:

```
A(Tp) = A(a(0), b(0)) = (a(32), b(32)) = Tc.
```

6. Режим гаммирования

6.1. Зашифровка открытых данных в режиме гаммирования

Открытые данные, разбитые на блоки по 64 бита Tp(1), Tp(2), ..., Tp(M-1), Tp(M), зашифровываются в режиме гаммирования путём поразрядного суммирования по модулю 2 в сумматоре СМ5 рабочим ключом (гаммой шифра) Gc, который генерируется блоками по 64 бита:

```
Gc = (Gc(1), Gc(2), ..., Gc(M-1), Gc(M))
```

где M определяется размером шифруемых данных.

Gc(i) - это i-й блок данных размером 64 бита, где i=1..M - число двоичных разрядов в блоке Tp(M) - может быть меньше 64 (в этом случае неиспользованная для шифрования часть рабочего ключа Gc(M) отбрасывается).

256 битов ключа помещаются в KDS. В регистры N1 и N2 помещаются 64-битовая двоичная последовательность (вектор инициализации или синхропосылка) S = (S1, S2, ..., S64), которая служит исходными данными для генерации M блоков рабочего ключа. Вектор инициализации помещается в регистры N1 и N2 следующим образом:

- значение S1 записывается в первый бит регистра N1;
- значение S2 записывается во второй бит регистра N1;
- ...
- значение S32 записывается в 32-й бит регистра N1;
- значение S33 записывается в первый бит регистра N2;
- значение S34 записывается во второй[2] бит регистра N2;
- ...
- значение S64 записывается в 32-й бит регистра N2.

Исходное содержимое регистров N1 и N2 (вектор инициализации S) шифруется в режиме простой замены в соответствии с требованиями параграфа 5.1. Результат шифрования A(S) = (Y0, Z0) записывается в 32-битовые регистры N3 и N4 (содержимое N1 записывается в N3, содержимое N2 - в N4).

[1] В оригинале это предложение содержит ошибку. См. https://www.rfc-editor.org/errata_search.php?eid=2140. *Прим. перев.*
[2] В оригинале ошибочно указан 33-й бит. См. https://www.rfc-editor.org/errata_search.php?eid=2094. *Прим. перев.*

Содержимое регистра N4 суммируется по модулю (2^{32}-1) в сумматоре CM4 с 32-битовой константой C1 из регистра N6, а результат записывается в регистр N4. Содержимое регистра N3 складывается по модулю 2^{32} в сумматоре CM3 с 32-битовой константой C2 из регистра N5, а результат записывается в регистр N3.

Содержимое регистра N3 копируется в N1, а содержимое N4 - в N2, значения регистров N3 и N4 при этом не меняются.

Содержимое регистров N1 и N2 шифруется методом простой замены в соответствии с требованиями параграфа 5.1. Полученное в результате содержимое регистров N1 и N2 образует первый 64-битовый блок рабочего ключа Gc(1), который поразрядно суммируется по модулю 2 в сумматоре CM5 с первым 64-битовым блоком открытых данных:

```
Tp(1) = (t1(1), t2(1), ..., t63(1), t64(1)).
```

Результат суммирования даёт 64-битовый блок зашифрованных данных:

```
Tc(1) = (tau1(1), tau2(1), ..., tau63(1), tau64(1)).
```

Значение tau1(1) блока Tc(1) является результатом сложения по модулю 2 в сумматоре CM5 значения t1(1) из блока Tp(1) со значением первого бита регистра N1, tau2(1) блока Tc(1) - результатом сложения по модулю 2 в сумматоре CM5 значения t2(1) из блока Tp(1) со значением второго бита N1 и т. д., значение tau64(1) блока Tc(1) является результатом сложения по модулю 2 в сумматоре CM5 значения t64(1) из блока Tp(1) со значением 32-го бита N2.

Для получения следующего 64-битового блока рабочего ключа Gc(2) содержимое регистра N4 складывается по модулю (2^{32}-1) в сумматоре CM4 с константой C1 из регистра N6; содержимое регистра N3 складывается по модулю 2^{32} в сумматоре CM3 с константой C2 из регистра N5. Новое значение регистра N3 копируется в N1, новое значение N4 - в N2[1]. Значения регистров N3 и N4 при этом сохраняются.

Содержимое регистров N1 и N2 шифруется методом простой замены в соответствии с требованиями параграфа 5.1. Полученное в результате зашифрованное содержимое регистров N1 и N2 образует первый 64-битовый блок рабочего ключа Gc(2), который поразрядно суммируется по модулю 2 в сумматоре CM5 со вторым[2] 64-битовым блоком открытых данных Tp(2). Генерация остальных блоков рабочего ключа Gc(3), Gc(4), ..., Gc(M) и зашифровка открытых блоков Tp(3), Tp(4), ..., Tp(M) выполняется аналогично. Если размер последнего (M-го) блока открытых данных менее 64 битов, из последнего блока рабочего ключа используется только соответствующее количество битов, а остальные биты отбрасываются[3].

Вектор инициализации S и блоки зашифрованных данных Tc(1), Tc(2), ..., Tc(M) передаются в канал связи или память ЭВМ.

Уравнение зашифровки имеет вид:

```
Tc(i) = A(Y[i-1] [+] C2, Z[i-1]) [+]' C1) (+) Tp(i) = Gc(i) (+) Tp(i)
i=1..M
```

где

Y[i] - содержимое регистра N3 после зашифровки i-го блока открытых данных Tp(i);

Z(i) - содержимое регистра N4 после зашифровки i-го блока открытых данных Tp(i);

```
(Y[0], Z[0]) = A(S).
```

6.2. Расшифровка данных с режиме гаммирования

В KDS вводятся 256 битов ключа, использованного для зашифровки открытых данных Tp(1), Tp(2), ..., Tp(M). Значение вектора инициализации S помещается в регистры N1 и N2 и генерируются M блоков рабочего ключа Gc(1), Gc(2), ..., Gc(M), как описано в параграфе 6.1. Блоки зашифрованных данных Tc(1), Tc(2), ..., Tc(M) поразрядно складываются по модулю 2 в сумматоре CM5 с блоками рабочего ключа, что приводит к восстановлению блоков открытых данных Tp(1), Tp(2), ..., Tp(M). Блок Tp(M) может иметь размер менее 64 битов.

Уравнение расшифровки имеет вид:

```
Tp(i) = A (Y[i-1] [+] C2, Z[i-1] [+]' C1) (+) Tc(i) = Gc(i) (+) Tc(i)
i = 1..M
```

7. Режим гаммирования с обратной связью

7.1. Зашифровка данных в режиме гаммирования с обратной связью

Открытые данные разбиваются на блоки размером 64 бита Tp(1), Tp(2), ..., Tp(M) и шифруются в режиме гаммирования с обратной связью путём поразрядного сложения по модулю в сумматоре CM5 с рабочим ключом (гаммой) Gc, генерируемым в форме 64-битовых блоков. Т. е., Gc(i)=(Gc(1), Gc(2), ..., Gc(M)), где M определяется размером открытых данных, Gc(i) - i-й блок ключа размером 64 бита, i=1..M. Размер блока Tp(M) может быть менее 64 битов[4].

В KDS вводятся 256 битов ключа. 64-битовый вектор инициализации (синхропосылка) S = (S1, S2, ..., S64) помещается в регистры N1 и N2, как описано в параграфе 6.1.

Исходное содержимое регистров N1 и N2 зашифровывается в режиме простой замены в соответствии с требованиями параграфа 5.1[5]. Зашифрованное содержимое регистров N1 и N2 является первым 64-битовым блоком рабочего ключа Gc(1)=A(S), который складывается по модулю 2 в сумматоре CM5 с первым 64-битовым блоком открытых данных Tp(1) = (t1(1), t2(1), ..., t64(1))[6].

---

[1] Здесь, равно как и в оригинальном стандарте, опущены операции копирования результатов суммирования в регистры N3 и N4, явно упомянутые для генерации первого блока рабочего ключа. *Прим. перев.*

[2] В оригинале ошибочно сказано «с первым». См. https://www.rfc-editor.org/errata_search.php?eid=2144. *Прим. перев.*

[3] Ни здесь, ни в оригинальном стандарте явно не сказано, что следует брать из M-го блока рабочего ключа начальные (старшие) биты. *Прим. перев.*

[4] В оригинале это предложение содержит ошибку. См. https://www.rfc-editor.org/errata_search.php?eid=2152. *Прим. перев.*

[5] В оригинале дана ссылка на параграф 6.1, однако в исходном стандарте ссылка даётся на пункт, соответствующий параграфу 5.1 данного документа. *Прим. перев.*

[6] В оригинале это предложение отличается от исходного стандарта и имеет вид: «Если зашифрованное содержимое регистров N1 и N2 является первым 64-битовым блоком рабочего ключа Gc(1)=A(S), этот блок складывается по модулю 2 с первым 64-битовым блоком открытых данных Tp(1) = (t1(1), t2(1), ..., t64(1))». Поскольку речь идёт о начале процесса зашифровки, в соответствии с

В результате получается 64-битовый блок зашифрованных данных

```
Tc(1) = (tau1(1), tau2(1), ..., tau64(1)).
```

Блок зашифрованных данных Tc(1) одновременно используется в качестве исходных значений регистров N1 и N2 для генерации второго блока рабочего ключа Gc(2) и по цепи обратной связи этот блок записывается в регистры следующим образом:

- значение tau1(1) записывается в первый бит регистра N1;
- значение tau2(1) записывается во второй бит регистра N1;
- ...
- значение tau32(1) записывается в 32-й бит регистра N1;
- значение tau33(1) записывается в первый бит регистра N2;
- значение tau34(1) записывается во второй бит регистра N2;
- ...
- значение tau64(1) записывается в 32-й бит регистра N2.

Содержимое регистров N1 и N2 зашифровывается в режиме простой замены в соответствии с требованиями параграфа 5.1[5]. Зашифрованное содержимое регистров N1 и N2 образует второй 64-битовый блок рабочего ключа Gc(2), который складывается по модулю 2 в сумматоре CM5 со вторым блоком открытых данных Tp(2).

Генерация последующих блоков рабочего ключа Gc(i) и шифрование соответствующих блоков открытых данных Tp(i) (i = 3..M) выполняется аналогично. Если размер последнего (M-го) блока открытых данных меньше 64 битов, используются только соответствующие биты M-го блока рабочего ключа Gc(M), а оставшиеся биты отбрасываются.

Уравнение шифрования в режиме гаммирования с обратной связью имеет вид:

```
|Tc(1) = A(S) (+) Tp(1) = Gc(1) (+) Tp(1)
|Tc(i) = A(Tc(i-1)) (+) Tp(i) = Gc(i) + Tp(i), i = 2..M.
```

Вектор инициализации (синхропосылка) S и блоки зашифрованных данных Tc(1), Tc(2), ..., Tc(M) предаются в канал связи или память ЭВМ.

7.2. Расшифровка в режиме гаммирования с обратной связью

В KDS помещаются 256 битов ключа, использованного для зашифровки блока открытых данных Tp(1), Tp(2), ..., Tp(M). Вектор инициализации (синхропосылка) S в регистры N1 и N2, как описано в параграфе 6.1. Исходное содержимое регистров N1 и N2 (вектор инициализации S) шифруется в режиме простой замены в соответствии с требованиями параграфа 5.1[5]. Зашифрованное содержимое регистров N1, N2 образует первый блок рабочего ключа Gc(1) = A(S), который суммируется поразрядно по модулю 2 в сумматоре CM5 с блоком зашифрованных данных Tc(1). Результатом этой операции является первый блок открытых (расшифрованных) данных Tp(1).

Блок зашифрованных данных Tc(1) служит исходным содержимым регистров N1, N2 при генерации второго блока рабочего ключа Gc(2). Блок Tc(1) записывается в регистры N1 и N2 в соответствии с требованиями параграфа 6.1. Содержимое регистров N1 и N2 шифруется в режиме простой замены в соответствии с требованиями параграфа 5.1 и полученный а результате блок Gc(2) поразрядно суммируется по модулю 2 в сумматоре CM5 со вторым блоком зашифрованных данных Tc(2). В результате получается второй блок расшифрованных данных Tc(2)[3].

Аналогично, блоки зашифрованных данных Tc(2), Tc(3), ..., Tc(M-1) поочерёдно записываются в регистры N1, N2, а содержимое этих регистров шифруется в режиме простой замены для генерации блоков рабочего ключа Gc(3), Gc(4), ..., Gc(M). Эти блоки поразрядно суммируются по модулю 2 в сумматоре CM5 с блоками зашифрованных данных Tc(3), Tc(4), ..., Tc(M), в результате чего получаются блоки открытых (расшифрованных) данных Tp(3), Tp(4), ..., Tp(M). Размер последнего блока открытых данных Tp(M) может быть менее 64 битов.

Уравнение расшифровки в режиме гаммирования с обратной связью имеет вид

```
Tp(1) = A(S) (+) Tc(1) = Gc(1) (+) Tc(1)
Tp(1) = A(Tc(i-1)) (+) Tc(i) = Gc(i) (+) Tc(i), i=2..M
```

8. Режим генерации MAC

Для защиты от фальсификации открытых данных, представляющих собой M блоков размером 64 бита Tp(1), Tp(2), ..., Tp(M), где M >= 2 генерируется дополнительный блок размером l (имитовставка или MAC - I(l)). Процесс генерации MAC в режимах зашифровки и расшифровки совпадает.

Первый блок открытых данных

```
Tp(1) = (t1(1), t1(2), ..., t64(1))
      = (a1(1)[0], a2(1)[0], ..., a32(1)[0], b1(1)[0], b2(1)[0], ..., b32(1)[0])
```

записывается в регистры N1 и N2 следующим образом:

- значение t1(1) = a1(1)[0] записывается в первый бит регистра N1;
- значение t2(1) = a2(1)[0] записывается во второй бит регистра N1;
- ...
- значение t32(1) = a32(1)[0] записывается в 32-й бит регистра N1;

предшествующим текстом описания этот блок может быть только первым и в переводе RFC был сохранен текст, более точно соответствующий исходному стандарту. См. также https://www.rfc-editor.org/errata_search.php?eid=2148. *Прим. перев.*

[3]В оригинале этот абзац содержит ошибку. См. https://www.rfc-editor.org/errata_search.php?eid=2135. *Прим. перев.*

- значение t33(1) = b1(1)[0] записывается в первый бит регистра N2;
- ...
- значение t64(1) = b32(1)[0] записывается в 32-й бит регистра N2.

Содержимое регистров N1 и N2 преобразуется в соответствии с первыми 16 циклами алгоритма шифрования в режиме простой замены (параграф 5.1[1]). В KDS при этом находится тот же ключ, который используется для зашифровки блоков открытых данных Tp(1), Tp(2), ..., Tp(M) в соответствующие блоки закрытых данных Tc(1), Tc(2), ..., Tc(M).

Полученное после 16 циклов содержимое регистров N1 и N2, имеющее вид (a1(1)[16], a2(1)[16], ..., a32(1)[16], b1(1)[16], b2(1)[16], ..., b32(1)[16]), суммируется по модулю 2 в сумматоре CM5 со вторым блоком открытых данных Tp(2) = (t1(2), t2(2), ..., t64(2)).

Результат суммирования

```
  (a1(1)[16](+)t1(2), a2(1)[16](+)t2(2), ..., a32(1)[16](+)t32(2),
 b1(1)[16](+)t33(2), b2(1)[16](+)t34(2), ..., b32(1)[16](+)t64(2))
=
  (a1(2)[0], a2(2)[0] ..., a32(2)[0], b1(2)[0], b2(2)[0], ..., b32(2)[0])
```

записывается в регистры N1, N2 и преобразуется в соответствии с первыми 16 циклами процесса шифрования в режиме простой замены.

Полученное в результате содержимое регистров N1 и N2 суммируется по модулю 2 в сумматоре CM5 с третьим блоком Tp(3) и т. д. Последний блок Tp(M) = (t1(M), t2(M), ..., t64(M)) при необходимости дополняется нулями до размера 64 бита и суммируется по модулю 2 в сумматоре CM5 с содержимым регистров N1 и N2[2]

```
(a1(M-1)[16], a2(M-1)[16], ..., a32(M-1)[16], b1(M-1)[16], b2(M-1)[16], ..., b32(M-1)[16]).
```

Результат суммирования

```
  (a1(M-1)[16](+)t1(M), a2(M-1)[16](+)t2(M), ..., a32(M-1)[16](+)
  t32(M), b1(M-1)[16](+)t33(M), b2(M-1)[16](+)t34(M), ..., b32(M-1)[16](+)t64(M))
=
  (a1(M)[0], a2(M)[0] ..., a32(M)[0], b1(M)[0], b2(M)[0], ..., b32(M)[0])
```

записывается в регистры N1, N2 и преобразуется в соответствии с первыми 16 циклами шифрования в режиме простой замены. Из полученного в результате содержимого регистров N1 и N2

```
(a1(M)[16], a2(M)[16] ..., a32(M)[16], b1(M)[16], b2(M)[16], ..., b32(M)[16])
```

выбирается отрезок размером l-битов - имитовставка (MAC) I(l):

```
I(l) = [a(32-l+1)(M)[16], a(32-l+2)(M)[16], ..., a32(M)[16]].
```

MAC I(l) передаётся по каналу связи или в память ЭВМ в конце зашифрованных данных. Т. е., результат имеет вид

```
Tc(1), Tc(2), ..., Tc(M), I(l).
```

Зашифрованные данные Tc(1), Tc(2), ..., Tc(M) расшифровываются, а из полученных при этом блоков открытых данных Tp(1), Tp(2), ..., Tp(M) генерируется MAC I'(l), как описано выше, и результат сравнивается с полученным (из канала передачи или памяти ЭВМ) вместе с зашифрованными данными значением MAC I(l). Если значения MAC не совпадают, полученные в результате расшифровки открытые данные Tp(1), Tp(2), ..., Tp(M) считаются фальсифицированы[3].

Значения MAC I(l), (I'(l)) могут генерироваться перед зашифровкой (после расшифровки) всего сообщения или параллельно с процессом зашифровки (расшифровки) блоков данных. Первые блоки открытых данных, используемые для генерации MAC, могут содержать служебную информацию (адресную часть, временную метку, вектор инициализации и т. п.) и не зашифровываться.

Значение параметра l (размер MAC) определяется действующими криптографическими требованиями с учётом того, что вероятность навязывания ложных данных равна 2^{-l}.

9. Вопросы безопасности

Документ целиком посвящён вопросам безопасности.

10. Нормативные документы

[GOST28147-89] «Система обработки информации. Защита криптографическая. Алгоритм криптографического преобразования», ГОСТ 28147-89, Государственный стандарт Союза ССР, Издательство стандартов, 1989.

[RFC4357] Popov, V., Kurepkin, I., and S. Leontiev, "Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms", RFC 4357, January 2006.

Приложение A. Значения констант C1 и C2

Константа C1 имеет вид:

Разряд N6	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Значение бита	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0

Константа C2 имеет вид:

Разряд N5[1]	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Значение бита	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

[1] В оригинале ошибочно указан параграф 6.1. См. https://www.rfc-editor.org/errata_search.php?eid=2151. *Прим. перев.*
[2] В оригинале этот абзац содержит ошибку. См. https://www.rfc-editor.org/errata_search.php?eid=2153. *Прим. перев.*
[3] В оригинале этот абзац содержит ошибку. См. https://www.rfc-editor.org/errata_search.php?eid=2154. *Прим. перев.*
[1] В оригинале ошибочно указан регистр N6. См. https://www.rfc-editor.org/errata_search.php?eid=2692. *Прим. перев.*

Приложение B. Разработчики документа

Dmitry Kabelev

Cryptocom, Ltd.

14 Kedrova St., Bldg. 2

Moscow, 117218

Russian Federation

EMail: kdb@cryptocom.ru

Igor Ustinov

Cryptocom, Ltd.

14 Kedrova St., Bldg. 2

Moscow, 117218

Russian Federation

EMail: igus@cryptocom.ru

Irene Emelianova

Cryptocom Ltd.

14 Kedrova St., Bldg. 2

Moscow, 117218

Russian Federation

EMail: irene@cryptocom.ru

Адрес автора

Vasily Dolmatov, редактор

Cryptocom, Ltd.

14 Kedrova St., Bldg. 2

Moscow, 117218

Russian Federation

EMail: dol@cryptocom.ru

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru