ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

РЕКОМЕНДАЦИИ ПО СТАНДАРТИЗАЦИИ

Р 1323565.1.029— 2019

Информационная технология

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Протокол защищенного обмена для индустриальных систем

Издание официальное

Москва
Стандартинформ
2020

Предисловие

1 РАЗРАБОТАНЫ Открытым акционерным обществом «Информационные технологии и коммуникационные системы» (ОАО «ИнфоТеКС»)

2 ВНЕСЕНЫ Техническим комитетом по стандартизации ТК 26 «Криптографическая защита информации»

3 УТВЕРЖДЕНЫ И ВВЕДЕНЫ В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 30 декабря 2019 г. № 1504-ст

4 ВВЕДЕНЫ ВПЕРВЫЕ

Правила применения настоящих рекомендаций установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящим рекомендациям публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящих рекомендаций соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.gost.ru)

© Стандартинформ, оформление, 2020

Настоящие рекомендации не могут быть полностью или частично воспроизведены, тиражированы и распространены в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

Введение

Настоящие рекомендации содержат описание протокола CRISP — CRyptographic Industrial Security Protocol — неинтерактивного протокола защищенной передачи данных, разработанного для применения в индустриальных системах. Протокол CRISP может быть использован для обеспечения конфиденциальности и имитозащиты сообщений и для защиты от навязывания повторных сообщений.

Протокол CRISP реализует защиту исходных сообщений путем их опционального шифрования, а также снабжения дополнительными данными, в частности, для обеспечения имитозащиты сообщений и для защиты от навязывания повторных сообщений с использованием криптографических методов.

Протокол CRISP не предназначен для встраивания в какой-либо определенный протокол передачи данных. Он представляет собой совокупность набора полей, правил их формирования и обработки. При этом на защищаемую систему возлагается задача доставки сформированных данных посредством используемых протоколов. В частности, адресация и маршрутизация данных возлагается на защищаемую систему.

Примечание —Настоящие рекомендации дополнены приложением А.

РЕКОМЕНДАЦИИ ПО СТАНДАРТИЗАЦИИ

Информационная технология

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Протокол защищенного обмена для индустриальных систем

Information technology. Cryptographic data security. Cryptographic industrial security protocol

Дата введения — 2020—09—01

1 Область применения

Областью применения протокола CRISP являются системы с жесткими ограничениями на длину передаваемых данных, требующие использования неинтерактивных протоколов.

2 Нормативные ссылки

В настоящих рекомендациях использованы нормативные ссылки на следующие стандарты:

ГОСТ Р 34.12—2015 Информационная технология. Криптографическая защита информации. Блочные шифры

ГОСТ Р 34.13—2015 Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров

Примечание — При пользовании настоящими рекомендациями целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящих рекомендаций в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, применяется в части, не затрагивающей эту ссылку.

3 Термины, определения и обозначения

3.1 Термины и определения

В настоящих рекомендациях применены следующие термины с соответствующими определениями:

3.1.1 **CRISP-сообщение:** Сообщение, защищенное с помощью протокола CRISP.

3.1.2 **базовый ключ:** Секретный ключ, известный только отправителю и получателю.

3.1.3 **идентификатор ключа:** Информация, использующаяся при определении ключа обработки CRISP-сообщения.

3.1.4 **исходное сообщение:** Сообщение до защиты его протоколом CRISP.

3.1.5 **криптографический набор:** Совокупность криптографических алгоритмов и параметров, используемых в протоколе CRISP.

Издание официальное

3.1.6 **окно принятых сообщений:** Диапазон допустимых порядковых номеров CRISP-сообщений, в котором помечены порядковые номера принятых CRISP-сообщений.

Примечание — Максимальным номером окна принятых сообщений является максимальный номер среди принятых CRISP-сообщений; минимальный номер окна принятых сообщений определяется максимальным номером окна принятых сообщений и размером окна принятых сообщений.

3.1.7 **отправитель:** Узел, создающий CRISP-сообщение из исходного сообщения.

3.1.8 **получатель:** Узел, восстанавливающий исходное сообщения из CRISP-сообщения.

3.2 Обозначения

В настоящих рекомендациях использованы следующие обозначения:

V^* — множество всех двоичных строк конечной длины, включая пустую строку;

$|x|$ — длина (число компонент) строки $x \in V^*$;

V_s — множество всех двоичных строк длины s, где s — целое неотрицательное число; нумерация подстрок и компонент строки осуществляется справа налево, начиная с нуля;

$x||y$ — конкатенация двоичных строк x и y из V^*, то есть строка из $V_{|x|+|y|}$, в которой левая подстрока из $V_{|x|}$ совпадает со строкой x, а правая подстрока из $V_{|y|}$ совпадает со строкой y;

V^l_8 — множество всех байтовых строк длины l; имеет место соответствие между элементами множеств V^l_8 и V_{8l}, задаваемое равенством $(a_{l-1},\ldots,a_1,a_0) = x_{8l-1}||\ldots||x_1||x_0$, где $a_{i-1} = x_{8l-1}||\ldots||x_{8l-7}||x_{8l-8},\ldots,$ $a_0 = x_7||\ldots||x_1||x_0$, $x_i \in V_1$, $i = 0,1,\ldots,8l-1$;

0^r — двоичная строка, состоящая из r нулей;

$\mathrm{LSB}_s(x): V^* \setminus \bigcup_{i=0}^{s-1} V_i \to V_s$ — отображение, ставящее в соответствие строке $z_{m-1}||\ldots||z_1||z_0$, $m \geq s$, строку $z_{s-1}||\ldots||z_1||z_0$, $z_i \in V_1$, $i = 0,1,\ldots, m-1$;

$\mathrm{MSB}_s(x): V^* \setminus \bigcup_{i=0}^{s-1} V_i \to V_s$ — отображение, ставящее в соответствие строке $z_{m-1}||\ldots||z_1||z_0$, $m \geq s$, строку $z_{m-1}||\ldots||z_{m-s+1}||z_{m-s}$, $z_i \in V_1$, $i = 0,1,\ldots,m-1$;

binary (*'string',l*) — представление символьной строки *string* , состоящей из m символов, в виде байтовой строки длины l, $l \geq m$, при котором сначала осуществляется посимвольный (с сохранением порядка следования символов) перевод исходной строки в байтовую строку $(a_{m-1},\ldots,a_1,a_0)$ длины m в соответствии с ASCII-представлением каждого символа, после чего в случае $l = m$ в качестве результата выдается байтовая строка $(a_{m-1},\ldots,a_1,a_0)$, а в случае $l > m$ — байтовая строка $(0x00,\ldots,0x00, a_{m-1},\ldots,a_1,a_0)$ длины l;

byte (X, l) — представление целого числа X, $0 \leq X \leq 2^{8l}-1$, в виде байтовой строки длины l, при котором соответствующая итоговой байтовой строке двоичная строка $x_{8l-1}||\ldots||x_1||x_0$, $x_i \in V_1$, $i = 0,1,\ldots,8l-1$ есть бинарное представление числа X, т. е. $X = x_0 + x_1 \cdot 2 + \ldots + x_{8l-1} \cdot 2^{8l-1}$;

K_{ENC} — ключ шифрования сообщения;

K_{MAC} — ключ имитозащиты сообщения;

K — базовый ключ;

Size — размер окна принятых сообщений.

4 Состав CRISP-сообщения

Здесь и далее названия полей сообщений выделены прямым полужирным шрифтом. При указании конкретного значения поля используют курсив.

Для записи чисел используется сетевой порядок байтов (Big-endian).

Перечень полей CRISP-сообщения приведен в таблице 1.

Таблица 1 — Перечень полей CRISP-сообщения

Номер поля	Наименование поля		Длина поля в битах
1	Заголовок	**ExternalKeyIdFlag**	1
2		**Version**	15
3		**CS**	8
4		**KeyId**	От 8 до 1024, конкретное значение определяется значением старших битов **KeyId**
5		**SeqNum**	48
6	**PayloadData**		Переменная
7	**ICV**		Переменная, определяется значением **CS**

4.1 ExternalKeyIdFlag

Признак необходимости внешней информации для однозначного определения ключа обработки входящего CRISP-сообщения. Длина поля — 1 бит.

ExternalKeyIdFlag = 0 означает, что ключ обработки входящего CRISP-сообщения однозначно определяется значением *KeyId. ExternalKeyIdFlag* = 1 означает, что для однозначного определения ключа обработки входящего CRISP-сообщения требуется дополнительная информация.

4.2 Version

Версия CRISP-сообщения. Беззнаковое целое число. Длина поля — 15 битов.

Текущий документ описывает CRISP-сообщение, для которого *Version* = 0.

4.3 CS

Идентификатор криптографического набора. Беззнаковое целое число. Длина поля — 8 битов.

Идентификатор определяет криптографический набор, используемый для создания CRISP-сообщения или восстановления исходного сообщения из CRISP-сообщения. Всего может существовать не более 256 различных криптонаборов, исходя из 8-битной длины поля **CS** CRISP-сообщения.

Список механизмов и параметров, определяемых и/или описываемых в криптографическом наборе, приведен в таблице 2.

Таблица 2 — Состав криптографического набора

Параметр	Описание	Правила задания	Назначение
EncryptionAlg	Алгоритм шифрования данных	Описание блочного шифра (или ссылка на такое описание); описание режима работы блочного шифра (или ссылка на такое описание), включая задание всех необходимых параметров	Алгоритм используется при шифровании сообщения
MACAlg	Алгоритм выработки имитовставки	Описание алгоритма (или ссылка на такое описание), включая задание всех необходимых параметров	Алгоритм используется при выработке имитовставки сообщения
MACLength	Длина имитовставки	Длина имитовставки задается в байтах	—
DeriveIV	Алгоритм формирования синхропосылки	Описание алгоритма (или ссылки на такое описание); алгоритм должен быть согласован со спецификациями шифров и режимами их работы	Алгоритм используется для формирования синхропосылки при шифровании сообщения

Окончание таблицы 2

Параметр	Описание	Правила задания	Назначение
DeriveKey	Алгоритмы выработки производных ключей из базового ключа	Описание алгоритмов, включая задание всех необходимых параметров	Алгоритмы используются для формирования производных ключей шифрования сообщения и производных ключей имитозащиты сообщения

4.4 KeyId

Идентификатор ключа. Двоичная строка.

KeyId = 1000 0000_2 означает, что поле **KeyId** не используется. В остальных случаях:

- если MSB_1 (*KeyId*) = 0, то длина поля **KeyId** составляет 1 байт и оставшиеся 7 битов содержат значение идентификатора ключа;
- если MSB_1 (*KeyId*) = 1, то оставшиеся 7 битов первого байта интерпретируются как беззнаковое целое число и определяют количество дополнительных байтов (от 1 до 127). Дополнительные байты содержат значение идентификатора ключа.

4.5 SeqNum

Порядковый номер сообщения. Беззнаковое целое число. Длина поля — 48 битов.

Для протокола CRISP версии 1.0 используется также для формирования синхропосылки алгоритма шифрования.

4.6 PayloadData

Исходное сообщение или зашифрованное исходное сообщение. Поле переменной длины.

Применение шифрования при обработке сообщения определяется использованным криптографическим набором (значением поля **CS**). Для конкретного значения *CS* шифрование либо используется при обработке любого сообщения, либо не используется при обработке любого сообщения.

4.7 ICV

Имитовставка. Двоичная строка. Длина поля определяется использованным криптографическим набором (значением поля **CS**). Для конкретного значения *CS* длина поля **ICV** может принимать только одно фиксированное значение.

Поле содержит значение имитовставки, рассчитанной для полей 1—6 CRISP-сообщения.

5 Ограничения

Максимальный размер CRISP-сообщения (суммарная длина всех полей CRISP-сообщения) — 2048 байтов.

Предполагается следующее:

- отправитель и получатель имеют общий базовый ключ;
- с каждым базовым ключом ассоциирован идентификатор ключа, который может быть использован при определении ключа обработки CRISP-сообщения;
- с каждым базовым ключом ассоциирован идентификатор отправителя *SourceIdentifier,* который может быть как внешней по отношению к *KeyId* информацией, так и частью *KeyId*; идентификатор отправителя является уникальным, т. е. у разных отправителей в системе *SourceIdentifier* различны;
- отправитель и получатель имеют общие криптографические наборы;
- задача определения базового ключа обработки сообщения отправителем и получателем находится за рамками протокола CRISP;
- для отправителя и получателя настроен размер окна принятых сообщений.

Примечания

1 Способ установки на отправителе и получателе общих базового ключа, его идентификатора, криптографических наборов и *SourceIdentifier* находится за рамками протокола CRISP.

2 Под определением базового ключа обработки сообщения отправителем или получателем понимается поиск нужного ключа среди установленных на отправителе или получателе на основании *KeyId* и, при необходимости, дополнительной информации, например *SourceIdentifier*.

3 Размер идентификатора отправителя *SourceIdentifier* должен быть в пределах от 4 до 32 байтов.

4 Размер окна принятых сообщений *Size* должен быть в пределах от 1 до 256 (включительно) сообщений.

6 Обработка CRISP-сообщения

6.1 Инициализация порядкового номера сообщения и окна принятых сообщений

Перед первым использованием конкретного базового ключа с целью защиты сообщений отправитель устанавливает начальное значение (инициализирует) *SeqNum*. Инициализация *SeqNum* может потребоваться также для восстановления после сбоев в нумерации сообщений.

Настоящие рекомендации не специфицируют конкретный алгоритм инициализации *SeqNum*. При этом алгоритм инициализации совместно с правилом формирования порядкового номера исходящих сообщений должны обеспечивать нахождение значения *SeqNum* в диапазоне от 0 до $2^{48} - 1$ (включительно) и строгое возрастание значения *SeqNum* для каждого сообщения, создаваемого одним отправителем с использованием одного базового ключа.

Перед первым использованием конкретного базового ключа с целью восстановления сообщений получатель устанавливает максимальный и минимальный номера окна принятых сообщений равными 0.

6.2 Защита исходного сообщения отправителем

Предполагается, что определен базовый ключ, его идентификатор и криптонабор для формирования CRISP-сообщения данному получателю.

Для создания CRISP-сообщения выполняют следующую последовательность действий:

а) формируют порядковый номер сообщения *SeqNum* — текущее значение *SeqNum* увеличивают на 1;

б) из базового ключа вырабатывают ключи шифрования (в случае, если криптографическим набором предусмотрено шифрование) и имитозащиты;

в) формируют поля заголовка CRISP-сообщения — поля 1—5 таблицы 1;

г) если криптографическим набором предусмотрено шифрование, то зашифровывают исходное сообщение;

д) вычисляют значение имитовставки *ICV* для заголовка CRISP-сообщения и исходного сообщения (в случае, если криптографическим набором не предусмотрено шифрование) или заголовка CRISP-сообщения и зашифрованного сообщения (в случае, если криптографическим набором предусмотрено шифрование) — поля 1—6 таблицы 1.

6.3 Восстановление исходного сообщения получателем

Для восстановления исходного сообщения из CRISP-сообщения выполняют следующую последовательность действий:

а) если версия протокола CRISP, указанная в заголовке CRISP-сообщения, не поддерживается получателем, то сообщение блокируют;

б) согласно значению *KeyId* и, в случае *ExternalKeyIdFlag* = 1, дополнительной информации определяют базовый ключ;

в) проверяют допустимость значения *SeqNum* входящего CRISP-сообщения:

- если значение *SeqNum* входящего CRISP-сообщения меньше минимального номера окна принятых сообщений, то CRISP-сообщение блокируют и процедуру восстановления исходного сообщения прекращают;
- если значение *SeqNum* входящего CRISP-сообщения принадлежит окну принятых сообщений и данный порядковый номер CRISP-сообщения помечен как принятый в окне принятых сообщений, то CRISP-сообщение блокируют и процедуру восстановления исходного сообщения прекращают.

В остальных случаях значение *SeqNum* входящего CRISP-сообщения является допустимым и осуществляют переход к следующему пункту;

г) из базового ключа вырабатывают ключи шифрования (если криптографическим набором предусмотрено шифрование) и имитозащиты;

д) выполняют контроль целостности CRISP-сообщения путем проверки имитовставки. Если имитовставка не верна, то сообщение блокируют и процедуру восстановления исходного сообщения прекращают; если имитовставка верна, то переходят к следующему пункту;

е) обновляют окно принятых сообщений:

- если значение *SeqNum* входящего CRISP-сообщения принадлежит окну принятых сообщений, то порядковый номер *SeqNum* помечают в окне принятых сообщений как принятый;
- если значение *SeqNum* входящего CRISP-сообщения больше максимального номера окна принятых сообщений, то новым максимальным номером окна принятых сообщений становится значение

SeqNum, порядковый номер *SeqNum* помечают в окне принятых сообщений как принятый, а новым минимальным номером окна принятых сообщений становится значение *SeqNum — Size* + 1 или 0, если значение *SeqNum — Size* + 1 меньше 0;

ж) если криптографическим набором предусмотрено шифрование, то выполняется расшифрование значения поля **PayloadData**, восстанавливают исходное сообщение.

7 Криптографические наборы

7.1 Набор MAGMA-CTR-CMAC: CS=1

Таблица 3 — Описание криптографического набора MAGMA-CTR-CMAC

Параметр	Значение
EncryptionAlg	Блочный шифр «Магма» согласно ГОСТ Р 34.12—2015 в режиме гаммирования согласно ГОСТ Р 34.13—2015
MACAlg	Блочный шифр «Магма» согласно ГОСТ Р 34.12—2015 в режиме выработки имитовставки согласно ГОСТ Р 34.13—2015
MACLength	4 байта
DeriveIV	См. описание далее
DeriveKey	См. описание далее

7.1.1 Алгоритм шифрования

Для шифрования данных используют блочный шифр «Магма» согласно ГОСТ Р 34.12—2015 в режиме гаммирования согласно ГОСТ Р 34.13—2015. В качестве ключа используют ключ шифрования сообщения K_{ENC}. Значение входного параметра режима гаммирования $s = 64$. В качестве синхропосылки используют значение, определенное в 7.1.3. Криптографическое дополнение данных для режима гаммирования не предусмотрено.

7.1.2 Имитовставка

Для вычисления имитовставки *ICV,* содержащейся в поле **ICV** CRISP-сообщения, используют блочный шифр «Магма» согласно ГОСТ Р 34.12—2015 в режиме выработки имитовставки согласно ГОСТ Р 34.13—2015. В качестве ключа используют ключ имитозащиты сообщения K_{MAC}. Значение входного параметра режима выработки имитовставки $s = 32$. Криптографическое дополнение данных для режима выработки имитовставки выполняют согласно ГОСТ Р 34.13—2015.

7.1.3 Синхропосылка

Для формирования из 48-битного порядкового номера сообщения *SeqNum,* содержащегося в поле **SeqNum** CRISP-сообщения, 32-битной синхропосылки *IV* используют 32 младших бита *SeqNum*:

$$IV = LSB_{32}(byte(SeqNum, 6)).$$

7.1.4 Ключи

Для выработки производных ключей шифрования и имитозащиты сообщения используют функцию

$$CMAC\ (Key, Data),$$

которую реализуют с помощью блочного шифра «Магма» согласно ГОСТ Р 34.12—2015 в режиме выработки имитовставки согласно ГОСТ Р 34.13—2015 для данных *Data* на ключе *Key*. Значение входного параметра режима выработки имитовставки $s = 64$. Криптографическое дополнение для режима выработки имитовставки выполняют согласно ГОСТ Р 34.13—2015.

Для выработки производного ключа шифрования сообщения и производного ключа имитозащиты сообщения вычисляют следующие величины:

- *Key* — инициализируют базовым ключом *K*;
- *Label* = binary ('macenc', 6);
- *aL* = byte (6,1);
- $SN = 0^5 || MSB_{35}$ (byte(*SeqNum*, 6)), где

SeqNum инициализируют значением *SeqNum* CRISP-сообщения;

- *Node* = *SourceIdentifier*, где

SourceIdentifier инициализируют значением идентификатора отправителя;

- *CS* — инициализируют значением *CS* CRISP-сообщения;
- *cL* = byte(*ContextLength*, 2), где

ContextLength — сумма байтовых длин значений *SN*, *Node* и *CS*;
- *OutputLength* = 512, где
OutputLength — необходимая битовая длина вырабатываемого ключевого материала;
- *oL* = byte (*OutputLength*, 2).
Для каждого числа i = 1,…,8 вычисляют 64-битные величины:
K_i = CMAC (*Key,byte*(*i*,1)||*Label*||*aL*||*SN*||*Node*||byte(*CS*,1)||*cL*||*oL*).
Ключи шифрования и имитозащиты сообщения вычисляют как

$$K_{MAC} = K_1||K_2||K_3||K_4;$$

$$K_{ENC} = K_5||K_6||K_7||K_8.$$

7.2 Набор MAGMA-NULL-CMAC: CS=2

Таблица 4 — Описание криптографического набора MAGMA-NULL-CMAC

Параметр	Значение
EncryptionAlg	Не используется
MACAlg	Блочный шифр «Магма» согласно ГОСТ Р 34.12—2015 в режиме выработки имитовставки согласно ГОСТ Р 34.13—2015
MACLength	4 байта
DeriveIV	Не используется
DeriveKey	См. описание далее

7.2.1 Имитовставка

Для вычисления имитовставки *ICV*, содержащейся в поле **ICV** CRISP-сообщения, используют блочный шифр «Магма» согласно ГОСТ Р 34.12—2015 в режиме выработки имитовставки согласно ГОСТ Р 34.13—2015. В качестве ключа используют ключ имитозащиты сообщения K_{MAC}. Значение входного параметра режима выработки имитовставки s = 32. Криптографическое дополнение данных для режима выработки имитовставки выполняют согласно ГОСТ Р 34.13—2015.

7.2.2 Ключ

Для выработки производного ключа имитозащиты сообщения используют функцию

CMAC (*Key,Data*),

которую реализуют с помощью блочного шифра «Магма» согласно ГОСТ Р 34.12—2015 в режиме выработки имитовставки согласно ГОСТ Р 34.13—2015 для данных *Data* на ключе *Key*. Значение входного параметра режима выработки имитовставки s = 64. Криптографическое дополнение для режима выработки имитовставки выполняют согласно ГОСТ Р 34.13—2015.

Для выработки производного ключа имитозащиты сообщения вычисляют следующие величины:
- *Key* — инициализируют базовым ключом *K*;
- *Label* = binary ('macmac', 6);
- *aL* = byte (6,1);
- *SN* = 0^5||MSB_{35}(*byte*(*SeqNum*, 6)), где
SeqNum инициализируют значением *SeqNum* CRISP-сообщения;
- *Node* = *SourceIdentifier*, где
SourceIdentifier инициализируют значением идентификатора отправителя;
- *CS* — инициализируют значением *CS* CRISP-сообщения;
- *cL* = byte(*ContextLength*, 2), где
ContextLength — сумма байтовых длин значений *SN*, *Node* и *CS*;
- *OutputLength* = 256, где
OutputLength — необходимая битовая длина вырабатываемого ключевого материала;
- *oL* = byte (*OutputLength*, 2).
Для каждого числа i = 1,…,4 вычисляют 64-битные величины:

K_i = CMAC (*Key*,byte(*i*, 1)||*Label*||*aL*||*SN*||*Node*||byte(*CS*, 1)||*cL*||*oL*).

Ключ имитозащиты сообщения вычисляют как

$$K_{MAC} = K_1||K_2||K_3||K_4.$$

Приложение А
(справочное)

Контрольные примеры

Приводимые ниже значения *SourseIdentifier, KeyId, SeqNum* и базового ключа *K* рекомендуется использовать только для проверки корректной работы конкретной реализации алгоритмов, описанных в настоящих рекомендациях.

Все числовые значения приведены в шестнадцатеричной системе счисления.

В данном приложении двоичные строки из V^*, длина которых кратна 4, записываются в шестнадцатеричном виде, а символ конкатенации («||») опускается. То есть строка $a \in V_{4r}$ будет представлена в виде $a_{r-1}a_{r-2}\ldots a_0$, где $a_i \in \{0,1,\ldots,9, a, b, c, d, e, f\}$, $i = 0,1,\ldots, r-1$. Соответствие между двоичными строками длины 4 и шестнадцатеричными строками длины 1 задается естественным образом (таблица А.1).

Преобразование, ставящее в соответствие двоичной строке длины $4r$ шестнадцатеричную строку длины r, и соответствующее обратное преобразование для простоты записи опускается.

Таблица А.1 — Соответствие между двоичными и шестнадцатеричными строками

Двоичная запись	Шестнадцатеричная запись
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

Примечание — Символ «\\», приведенный в А.1 и А.2, обозначает перенос числа на новую строку.

А.1 Набор MAGMA-CTR-CMAC: CS = 1

Для формирования сообщения используются следующие значения:
ExternalKeyIdFlag = 1
Version = 0
CS = 01_{16}
KeyId = 30_{16}
SeqNum = $0b76e6736001_{16}$
SourseIdentifier = $3032303531383030303031_{16}$:
K = 56509427153249653498524659324653\\
$04532945346593845073249576351290_{16}$
Исходное сообщение *PayloadData*:
48692120546869732069732074657374\\
20666f72204352495350206d65737361\\
$6765730a03_{16}$

На основе исходных данных получаются следующие значения ключа имитозащиты K_{MAC}, ключа шифрования K_{ENC}, зашифрованного сообщения и имитовставки *ICV*:

K_{MAC} = *eeb0f6814257ad08964eabe5e0993d38*\\
b2afc2ada24e8362d455db06951f2d93$_{16}$
K_{ENC} = *e3316ad28c788c38dafdeb9388e234bd*\\
30e5c901eeeb1788cdc1ec5db315e1a7$_{16}$
ICV = *887f0a32*$_{16}$
Зашифрованное сообщение:
d324643aefd97b93b18d343a2fba477e\\
c704cd8d14ac1cf74ceb25577af8fc2c\\
25fa9050a1$_{16}$
Итоговое сообщение будет иметь следующий вид:
800001300b76e6736001\\
d324643aefd97b93b18d343a2fba477e\\
c704cd8d14ac1cf74ceb25577af8fc2c\\
25fa9050a1\\
887f0a32$_{16}$

А.2 Набор MAGMA-NULL-CMAC: CS = 2

Для формирования сообщения используются следующие значения:
ExternalKeyIdFlag = 1
Version = 0
CS = 02_{16}
KeyId = 30_{16}
SeqNum = *0b76e66ea001*$_{16}$
SourseIdentifier = $303230353138303030303031_{16}$
K = 56509427153249653498524659324653\\
045329453465938450732495763512 90$_{16}$
Исходное сообщение *PayloadData*:
48692120546869732069732074657374\\
20666f72204352495350206d65737361\\
6765730a03$_{16}$
На основе исходных данных получаются следующие значения ключа имитозащиты K_{MAC} и имитовставки *ICV*:
K_{MAC} = *c3e3780f87f2caf539fdad56d9cb0340*\\
b1052c0ae8272ddc9601c921f81a7ca5b$_{16}$
ICV = *b97ade94*$_{16}$
Итоговое сообщение будет иметь следующий вид:
800002300b76e66ea001\\
48692120546869732069732074657374\\
20666f72204352495350206d65737361\\
6765730a03\\
b97ade94$_{16}$

УДК 681.3.06:006.354 ОКС 35. 040

Ключевые слова: криптографические протоколы, аутентификация, шифрование, защита от навязывания повторных сообщений, ключ

БЗ 1—2020/59

Редактор *Н.В. Таланова*
Технический редактор *В.Н. Прусакова*
Корректор *О.В. Лазарева*
Компьютерная верстка *Е.О. Асташина*

Сдано в набор 14.01.2020. Подписано в печать 20.02.2020. Формат 60×84 1/8. Гарнитура Ариал.
Усл. печ. л. 1,86. Уч.-изд. л. 1,68.
Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении во ФГУП «СТАНДАРТИНФОРМ» для комплектования Федерального информационного фонда стандартов, 117418 Москва, Нахимовский пр-т, д. 31, к. 2.
www.gostinfo.ru info@gostinfo.ru

P1223SG1.029-2019