ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ
(РОССТАНДАРТ)

Технический комитет 026

«КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ»

СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ
ЗАЩИТА КРИПТОГРАФИЧЕСКАЯ

РЕКОМЕНДАЦИИ ПО СТАНДАРТИЗАЦИИ

ИСПОЛЬЗОВАНИЕ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ, СОПУТСТВУЮЩИХ ПРИМЕНЕНИЮ СТАНДАРТОВ ГОСТ Р 34.10-2012 И ГОСТ Р 34.11-2012

Утверждены решением заседания
технического комитета по стандартизации
«Криптографическая защита информации»
(Протокол №13 от 24.04.2014 г.)

Москва
2014

Содержание

1 Введение

Использование криптографических алгоритмов, определённых стандартами ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012, в средствах защиты информации осуществляется, как правило, в рамках криптографических протоколов, базирующихся на сопутствующих алгоритмах.

Спецификации предлагаемых в настоящем документе алгоритмов и параметров определены на основе опыта построения криптографических протоколов, отраженных в документах RFC4357, RFC4490 и RFC4491.

Настоящие рекомендации содержат описания сопутствующих алгоритмов, предназначенных для определения псевдослучайных функций протоколов, функций преобразования ключей, согласования ключей по протоколу Диффи-Хеллмана и экспорта ключевого материала.

Настоящая рекомендация не определяет криптографические алгоритмы ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012. Алгоритмы определены национальными стандартами ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012.

Необходимость разработки настоящего документа вызвана потребностью в обеспечении совместимости криптографических протоколов различных производителей, использующих алгоритмы ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012.

2 Область применения

Настоящий документ рекомендуется применять в системах шифрования и защиты аутентичности данных, использующих алгоритмы стандартов электронной подписи ГОСТ Р 34.10-2012 и функции хэширования ГОСТ Р 34.11-2012, в общедоступных и корпоративных сетях для защиты информации, не содержащей сведений, составляющих государственную тайну.

3 Нормативные ссылки

Указанные в этом разделе рекомендаций ссылочные документы являются обязательными для их применения. Для датированных ссылок используют только указанное здесь издание. Для недатированных ссылок - последнее и актуальное издание со всеми изменениями и дополнениями:

ГОСТ Р 34.10-2012 — Федеральное Агентство по техническому регулированию и метрологии. Национальный стандарт Российской Федерации. Информационная технология. Криптографическая защита информации. «Процессы формирования и проверки электронной цифровой подписи». Стандартинформ, Москва 2013.

ГОСТ Р 34.11-2012 — Федеральное Агентство по техническому регулированию и метрологии. Национальный стандарт Российской Федерации. Информационная технология. Криптографическая защита информации. «Функция хэширования». Стандартинформ, Москва 2013.

3.1 Дополнительные ссылки

Указанные в этом разделе рекомендаций ссылочные документы не являются обязательными для их применения в Российской Федерации согласно установленным в настоящее время правилам подготовки нормативных документов Росстандарта. Тем не менее, следует учитывать сложившуюся практику использования документов IETF в качестве международных предстандартов, признаваемых и используемых нормативно разработчиками интернет-технологий всех технологически развитых стран.

RFC2104 — Х.Кравчик, М. Белларе и Р. Канетти, «HMAC: ключевое хэширование для проверки подлинности сообщений» (H. Krawczyk, M. Bellare and R. Canetti, HMAC: Keyed-Hashing for Message Authentication, IETF RFC 2104, February 1997).

RFC2119 — С. Браднер, «Ключевые слова для использования в документах RFC, указывающие уровень требований», стандарт BCP 14, март 1997 г. (Bradner S., Key words for use in RFCs to Indicate Requirement Levels, BCP 14, IETF RFC 2119, March 1997).

RFC2246 — К. Аллен, Т. Диркс, «Протокол TLS – Transport Layer Security – версия 1.0» (C. Allen, T. Dierks, The TLS Protocol Version 1.0, IETF RFC 2246, January 1999).

RFC2409 — Д. Харкинс, Д. Каррел, «Протокол обмена ключами в сети Интернет (IKE)» (Harkins, D. and D. Carrel, The Internet Key Exchange (IKE), IETF RFC 2409, November 1998).

RFC4357 — В. Попов, И. Курепкин, С. Леонтьев, «Дополнительные алгоритмы шифрования для использования с алгоритмами по ГОСТ 28147-89, ГОСТ Р 34.10-94, ГОСТ Р 34.10-2001 и ГОСТ Р 34.11-94» (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, IETF RFC 4357, January 2006).

RFC5246 — Т. Диркс, Е. Рескорла, «Протокол TLS – Transport Layer Security – версия 1.0» (T. Dierks, E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.2, IETF RFC 5246, August 2008).

RFC5996 — С. Кауфман, П. Хофман, Й. Нир, П. Еронен, «Протокол обмена ключами в сети Интернет, версия 2 (IKEv2)» (Kaufman C., Hoffman P., Nir Y., and P. Eronen, Internet Key Exchange Protocol Version 2 (IKEv2), IETF RFC 5996, September 2010).

NIST SP 800-108 — Лили Чен, «Рекомендации по получению производных ключей с использованием псевдослучайных функций», Национальный институт стандартов и технологий США (Lily Chen, Recommendation for Key Derivation Using Pseudorandom Functions, National Institute of Standards and Technology (NIST) Special Publication 800-108, October 2009).

4 Обозначения, термины и сокращения

4.1 Обозначения математических объектов

В настоящем документе используются следующие обозначения для множеств и операций над элементами этих множеств в соответствии с ГОСТ Р 34.11-2012.

— V_n –конечномерное векторное пространство над GF(2) размерности n с операцией сложения $\oplus$, v – элемент V_n, в битовом представлении $v=(v_{n-1},v_{n-2},\ldots,v_1,v_0)$, $v_i \in \{0,1\}$, при $n = 0$ пространство V_0 состоит из единственного пустого элемента длины 0.

— Если $A \in V_{n1}$, $B \in V_{n2}$, $A=(a_{n1-1}, a_{n1-2}, \ldots, a_0)$, $B=(b_{n2-1}, b_{n2-2}, \ldots, b_0)$, то $A|B=(a_{n1-1}, a_{n1-2}, \ldots, a_0, b_{n2-1}, b_{n2-2}, \ldots, b_0) \in V_{n1+n2}$.

— V_8^r – множество байтовых строк длины r, w – элемент V_8^r, в байтовом представлении $w=(w^0, w^1, \ldots, w^{r-1})$, где $w^0,w^1,\ldots,w^{r-1} \in V_8$.

— Если $A \in V_8^{r1}$, $B \in V_8^{r2}$, $A=(a^0, a^1, \ldots, a^{r1-1})$, $B=(b^0, b^1, \ldots, b^{r2-1})$, то $A|B=(a^0, a^1, \ldots, a^{r1-1}, b^0, b^1, \ldots, b^{r2-1}) \in V_8^{r1+r2}$.

— Битовым представлением элемента $w=(w^0, w^1, \ldots, w^{r-1}) \in V_8^r$, где $w^0=(w_7,w_6,\ldots,w_0),\ldots,w^{r-1}=(w_{8r-1},w_{8r-2},\ldots,w_{8r-8}) \in V_8$, называется элемент $w=(w_{8r-1},w_{8r-2},\ldots,w_1,w_0) \in V_{8r}$.

— При значении n, кратном 8, байтовым представлением элемента $W=(w_{n-1},w_{n-2},\ldots,w_0) \in V_n$ будем называть элемент V_8^r, $r = n/8$, равный $W=(w^0, w^1, \ldots, w^{r-1})$, где $w^0,w^1,\ldots,w^{r-1} \in V_8$, $w^0=(w_7,w_6,\ldots,w_0)$, $w^1=(w_{15},w_{14},\ldots,w_8)$, …, $w^{r-1}=(w_{n-1},w_{n-2},\ldots,w_{n-8})$.

— K – ключ – произвольный элемент из V_n, если $K \in V_n$, то его длина (в битах) равна n, n может быть произвольным натуральным числом.

Примечание: Интерпретацию формул и их редакцию предлагается проводить в соответствии с введёнными определениями.

4.2 Основные понятия, термины и определения

В настоящем документе используются следующие понятия, аббревиатуры и обозначения:

H_{256}	хэш-функция ГОСТ Р 34.11-2012, 256 бит
H_{512}	хэш-функция ГОСТ Р 34.11-2012, 512 бит
HMAC	функция, предназначенная для вычисления кода аутентификации сообщения (основана на некоторой хэш-функции)
$HMAC_{256}$	функция, основанная на хэш-функции H_{256}, предназначенная для вычисления кода аутентификации сообщения
$HMAC_{512}$	функция, основанная на хэш-функции H_{512}, предназначенная для вычисления кода аутентификации сообщения
PRF	псевдослучайная функция — отображение, позволяющее вырабатывать псевдослучайные последовательности байтов
KDF	функция диверсификации — отображение, позволяющее порождать производные ключи и ключевой материал по корневому ключу и произвольным данным с использованием псевдослучайной функции

Для выработки байтовой последовательности длины N с помощью функций, вырабатывающих последовательности, вообще говоря, большей длины, необходимо взять из выходной последовательности N первых байтов. Это замечание относиться к следующим функциям, описанным в данном документе:

- Функции, описанные в разделе «Функции PRF»
- *KDF_TREE_GOSTR3411_2012_256*

Один и тот же элемент пространства V_n при n кратном 8 имеет битовое и байтовое представления. Результат операции «|», примененной к элементам в битовом представлении, описывается в битовом представлении. Результат операции «|», примененной к тем же элементам в их байтовом представлении, описывается в байтовом представлении. Таким образом, символом «|» обозначены две различные операции, зависящие от представления их аргументов. Выбор одной из операций однозначно определяется в зависимости от выбранного представления аргументов.

Далее все данные (элементы пространства V_n), если явно не указано иное, считаются приведенными в байтовом представлении. Операция «|» на аргументах функций, если явно не указано иное, выполняется над их байтовым представлением.

Если используемая функция определена вне настоящего документа (например, H_{256}) и её определение использует аргументы в битовом представлении, то подразумевается, что битовое представление аргументов формируется непосредственно перед вычислением функции (в частности, только после применения операции «|» к байтовому представлению аргументов).

Если в качестве аргумента определяемых ниже функций используется выходное значение другой функции, которая определена вне настоящего документа и имеет выходные значение в битовом представлении, то подразумевается, что выходное значение перед подстановкой в аргументы переводится в байтовое представление.

5 Описание алгоритмов

Возможные значения аргументов функций в представленных алгоритмах ограничиваются допустимостью их использования в качестве входных параметров преобразований и присваиваются в протоколах.

5.1 Функции HMAC

В настоящем разделе определены преобразования HMAC на основе ГОСТ Р 34.11-2012 с различными длинами выходных значений.

5.1.1 HMAC_GOSTR3411_2012_256

Является преобразованием *HMAC* на основе ГОСТ Р 34.11-2012, 256 бит. Имеет следующий идентификатор:

— id-tc26-hmac-gost-3411-12-256, «1.2.643.7.1.1.4.1».

Вычисление $HMAC_{256}(K,T)$ для данных T произвольной длины на ключе K длины n битов состоит в формировании байтовой строки K^* длины 64 байта и выполнении преобразований над K^* и данными T с использованием хэш-функции H_{256}.

Допускаются любые значения длины n из интервала от 256 до 512.

Для формирования ключа K^*: если $n<512$, положить строку K^* равной байтовому представлению битовой строки $K|A$, где $A = (0,0,\ldots,0) \in V_{512-n}$; если $n=512$, положить K^* равной байтовому представлению K. Значение $HMAC_{256}(K,T)$ определяется выражением:

$$HMAC_{256}(K,T) = H_{256}(K^* \oplus opad | H_{256}(K^* \oplus ipad | T)),$$

где в байтовом представлении

$$ipad = (0x36|0x36|\ldots|0x36) \in V_8^{64},$$

$$opad = (0x5C|0x5C|\ldots|0x5C) \in V_8^{64}.$$

Данный алгоритм использует H_{256} в качестве хэш-функции в конструкции *HMAC*, описанной в RFC2104. Указанный способ формирования ipad и opad также приведен в RFC2104. Длина выхода $HMAC_{256}$ в байтах равна 32, длина блока итерационной процедуры функции сжатия для H_{256} в байтах равна 64 (в обозначениях **RFC2104** соответственно L=32, B=64).

5.1.2 HMAC_GOSTR3411_2012_512

Является преобразованием *HMAC* на основе ГОСТ Р 34.11-2012, 512 бит. Имеет следующий идентификатор:

— id-tc26-hmac-gost-3411-12-512 , «1.2.643.7.1.1.4.2».

Вычисление $HMAC_{512}(K,T)$ для данных T произвольной длины на ключе K длины n бит состоит в формировании байтовой строки K^* длины 64 байта и выполнении преобразований над K^* и данными T с использованием хэш-функции H_{512}.

Допускаются любые значения длины n из интервала от 256 до 512. Рекомендуется использовать значение $n = 512$.

Для формирования ключа K^*: если $n<512$, положить строку K^* равной байтовому представлению битовой строки $K|A$, где $A = (0,0,\ldots,0) \in V_{512-n}$; если $n=512$, положить K^* равной байтовому представлению K. Значение $HMAC_{512}(K,T)$ определяется выражением:

$$HMAC_{512}(K,T) = H_{512}(K^* \oplus opad | H_{512}(K^* \oplus ipad | T)),$$

где в байтовом представлении

$ipad = (0x36|0x36|\ldots|0x36) \in V_8^{64}$,

$opad = (0x5C|0x5C|\ldots|0x5C) \in V_8^{64}$.

Данный алгоритм использует H_{512} в качестве хэш-функции в конструкции *HMAC*, описанной в **RFC2104**. Указанный способ формирования ipad и opad приведен также в **RFC2104**. Длина выхода $HMAC_{512}$ в байтах равна 64, длина блока итерационной процедуры функции сжатия для H_{512} в байтах равна 64 (в обозначениях **RFC2104** соответственно L=64, B=64).

5.2 Функции PRF

В настоящем разделе определены шесть рекомендуемых к использованию преобразований *PRF* – два для протокола TLS и четыре для протокола IPsec, построенные на основе *HMAC*.

Для получения с помощью любой из приведенных ниже *PRF* набора значений суммарной длины *m* байт следует положить их равными соответствующим последовательным значениям из первых *m* байт выхода используемой *PRF* в байтовом представлении.

5.2.1 Псевдослучайные функции протокола TLS

5.2.1.1 PRF_TLS_GOSTR3411_2012_256

Является преобразованием псевдослучайной функции протокола TLS, использующей $HMAC_{256}$ на основе ГОСТ Р 34.11-2012, 256 бит.

$$PRF\_TLS\_GOSTR3411\_2012\_256(secret, label, seed) = P\_GOSTR3411\_2012\_256\ (secret, label \mid seed)$$

$$P\_GOSTR3411\_2012\_256\ (secret, S) = HMAC_{256}\ (secret, A_1 \mid S) \mid HMAC_{256}\ (secret, A_2 \mid S) \mid HMAC_{256}\ (secret, A_3 \mid S) \mid \ldots$$

Параметры A_i определяются последовательно следующим образом:

$A_0 = S$

$A_i = HMAC_{256}\ (secret, A_{i-1})$

Функция *P_GOSTR3411_2012_25*6 использует $HMAC_{256}$ и соответствует способу задания аргументов и выходного значения функции расширения данных *P_hash*, приведенному в **RFC2246** (см. раздел 5) и сохраненному позднее в **RFC5246**.

5.2.1.2 PRF_TLS_GOSTR3411_2012_512

Является преобразованием псевдослучайной функции протокола TLS, использующей $HMAC_{512}$ на основе ГОСТ Р 34.11-2012, 512 бит.

$$PRF\_TLS\_GOSTR3411\_2012\_512(secret, label, seed) = P\_GOSTR3411\_2012\_512\ (secret, label \mid seed)$$

$$P\_GOSTR3411\_2012\_512\ (secret, S) = HMAC_{512}\ (secret, A_1 \mid S) \mid HMAC_{512}\ (secret, A_2 \mid S) \mid HMAC_{512}\ (secret, A_3 \mid S) \mid \ldots$$

Параметры A_i определяются последовательно следующим образом:

$A_0 = S$

$A_i = HMAC_{512}\ (secret, A_{i-1})$

Функция *P_GOSTR3411_2012_512* использует $HMAC_{512}$ и соответствует способу задания аргументов и выходного значения функции расширения данных *P_hash*, приведенному в **RFC2246** (см. раздел 5) и сохраненному позднее в **RFC5246**.

5.2.2 Псевдослучайные функции протокола IPsec на основе ГОСТ Р 34.11-2012, 256 бит

5.2.2.1 PRF_IPSEC_KEYMAT_GOSTR3411_2012_256

Псевдослучайная функция выработки ключевого материала определяется следующим образом (аргументами являются байтовые строки K и S):

$$PRF\_IPSEC\_KEYMAT\_GOSTR3411\_2012\_256\ (K,S) = T1 \mid T2 \mid T3 \mid T4 \mid \ldots,$$

где

$$T1 = HMAC_{256}\ (K,\ S)$$

$$T2 = HMAC_{256}\ (K,\ T1 \mid S)$$

$$T3 = HMAC_{256}\ (K,\ T2 \mid S)$$

$$T4 = HMAC_{256}\ (K,\ T3 \mid S)$$

$$\ldots$$

Функция *PRF_IPSEC_KEYMAT_GOSTR3411_2012_256* по схеме задания аргументов в итерациях аналогична функции *KEYMAT* в **RFC2409**.

5.2.2.2 PRF_IPSEC_PRFPLUS_GOSTR3411_2012_256

Псевдослучайная функция выработки ключевого материала определяется следующим образом (аргументами являются байтовые строки K и S):

$$PRF\_IPSEC\_PRFPLUS\_GOSTR3411\_2012\_256\ (K,S) = T1 \mid T2 \mid T3 \mid T4 \mid \ldots,$$

где

$$T1 = HMAC_{256}\ (K,\ S \mid 0x01)$$

$$T2 = HMAC_{256}\ (K,\ T1 \mid S \mid 0x02)$$

$$T3 = HMAC_{256}\ (K,\ T2 \mid S \mid 0x03)$$

$$T4 = HMAC_{256}\ (K,\ T3 \mid S \mid 0x04)$$

$$\ldots$$

Длина выхода *PRF_IPSEC_PRFPLUS_GOSTR3411_2012_256* не превышает 255·256 бит, что соответствует выходной последовательности *T1* | *T2* | *T3* | *T4* | *...* | *T255*.

Функция *PRF_IPSEC_PRFPLUS_GOSTR3411_2012_256* по схеме задания аргументов в итерациях аналогична функции *prf+* в **RFC5996**.

5.2.3 Псевдослучайные функции протокола IPsec на основе ГОСТ Р 34.11-2012, 512 бит

5.2.3.1 PRF_IPSEC_KEYMAT_GOSTR3411_2012_512

Псевдослучайная функция выработки ключевого материала определяется следующим образом (аргументами являются байтовые строки K и S):

PRF_IPSEC_KEYMAT_GOSTR3411_2012_512 (K,S) = T1 | T2 | T3 | T4 | ...,

где

$T1 = HMAC_{512}(K, S)$

$T2 = HMAC_{512}(K, T1 | S)$

$T3 = HMAC_{512}(K, T2 | S)$

$T4 = HMAC_{512}(K, T3 | S)$

...

Функция *PRF_IPSEC_KEYMAT_GOSTR3411_2012_512* по схеме задания аргументов в итерациях аналогична функции *KEYMAT* в **RFC2409**.

5.2.3.2 PRF_IPSEC_PRFPLUS_GOSTR3411_2012_512

Псевдослучайная функция выработки ключевого материала определяется следующим образом (аргументами являются байтовые строки K и S):

PRF_IPSEC_PRFPLUS_GOSTR3411_2012_512 (K,S) = T1 | T2 | T3 | T4 | ...,

где

$T1 = HMAC_{512}(K, S | 0x01)$

$T2 = HMAC_{512}(K, T1 | S | 0x02)$

$T3 = HMAC_{512}(K, T2 | S | 0x03)$

$T4 = HMAC_{512}(K, T3 | S | 0x04)$

...

Длина выхода *PRF_IPSEC_PRFPLUS_GOSTR3411_2012_512* не превышает 255·512 бит, что соответствует выходной последовательности *T1* | *T2* | *T3* | *T4* | *...* | *T255.*

Функция *PRF_IPSEC_PRFPLUS_GOSTR3411_2012_512* по схеме задания аргументов в итерациях аналогична функции *prf+* в **RFC5996**.

5.3 Алгоритмы согласования ключей VKO

В настоящем разделе определены алгоритмы согласования ключей с использованием GOST R 34.10-2012.

5.3.1 VKO_GOSTR3410_2012_256

Является алгоритмом согласования ключей VKO GOST R 34.10-2012, VKO 256 бит, на основе ГОСТ Р 34.11-2012, 256 бит. Может использоваться для согласования ключей ГОСТ Р 34.10-2012, 256 бит, а также ключей ГОСТ Р 34.10-2012, 512 бит.

Данный алгоритм предназначен для получения ключа шифрования либо ключевого материала длины 256 бит, далее используемых в криптографических протоколах. Ключ либо ключевой материал, обозначаемые $KEK_{VKO}(x,y,UKM)$, вырабатываются стороной обмена из своего закрытого ключа x, открытого ключа $y \cdot P$ противоположной стороны и величины *UKM*, рассматриваемой как число.

Алгоритм может использоваться как для статических, так и для эфемерных ключей сторон при битовой длине открытого ключа n, $n \geq 512$, в том числе и для случая, когда ключи одной из сторон являются статическими, а другой – эфемерными.

UKM используется опционально (в противном случае далее полагается *UKM*=1) и принимает значение от 1 до $2^{n/2}$-1. Допускается осуществлять выбор ненулевого значения *UKM* любой битовой длины, не превосходящей n/2. Использование *UKM* с битовой длиной не менее 64 рекомендуется в случае, когда ключи хотя бы одной из сторон являются статическими.

$$K(x,y,UKM) = (m/q \cdot UKM \cdot x \bmod q) \cdot (y \cdot P),$$

где *m* и *q* – параметры используемой эллиптической кривой, соответствующие обозначениям, принятым в ГОСТ Р 34.10-2012.

$$KEK_{VKO}(x,y,UKM) = H_{256}(K(x,y,UKM))$$

Данный алгоритм определяется по аналогии с разделом 5.2 в **RFC4357** при использовании вместо хэш-функции **ГОСТ Р 34.11-94** (обозначена в документе как gostR3411) хэш-функции H_{256} и вычислении *K(x,y,UKM)* при длинах открытых ключей *n≥512* бит и длине *UKM* – до *n/2* бит.

5.3.2 VKO_GOSTR3410_2012_512

Является алгоритмом согласования ключей VKO GOST R 34.10-2012, VKO 512 бит, на основе ГОСТ Р 34.11-2012, 512 бит. Может использоваться для согласования ключей ГОСТ Р 34.10-2012, 512 бит.

Данный алгоритм предназначен для получения ключа шифрования либо ключевого материала длины 512 бит, далее используемых в криптографических протоколах. Ключ либо ключевой материал, обозначаемые $KEK_{VKO}(x,y,UKM)$, вырабатываются стороной обмена из своего закрытого ключа *x*, открытого ключа $y \cdot P$ противоположной стороны и величины *UKM*, рассматриваемой как число.

Алгоритм может использоваться как для статических, так и для эфемерных ключей сторон при битовой длине открытого ключа *n, n≥1024*, в том числе и для случая, когда ключи одной из сторон являются статическими, а другой – эфемерными.

UKM используется опционально (в противном случае далее полагается *UKM*=1) и принимает значение от 1 до $2^{n/2}$-1. Допускается осуществлять выбор ненулевого значения *UKM* любой битовой длины, не превосходящей n/2. Использование *UKM* с битовой длиной не менее 128 рекомендуется в случае, когда ключи хотя бы одной из сторон являются статическими.

$$K(x,y,UKM) = (m/q \cdot UKM \cdot x \bmod q) \cdot (y \cdot P),$$

где *m* и *q* – параметры используемой эллиптической кривой, соответствующие обозначениям, принятым в ГОСТ Р 34.10-2012.

$$KEK_{VKO}(x,y,UKM) = H_{512}(K(x,y,UKM))$$

Данный алгоритм определяется по аналогии с разделом 5.2 в **RFC4357** при использовании вместо хэш-функции ГОСТ Р 34.11-94 (обозначена в документе как gostR3411) хэш-функции H_{512} и вычислении K(x,y,*UKM*) при длинах открытых ключей n≥1024 бита и длине *UKM* – до n/2 бит.

5.4 Функция диверсификации KDF_GOSTR3411_2012_256

Функция диверсификации KDF_GOSTR3411_2012_256 на основе $HMAC_{256}$ предназначена для порождения ключевого материала длиной 256 бит и определяется выражением:

$$KDF(K_{in},label,seed) = HMAC_{256}(K_{in},0x01|label|0x00|seed|0x01|0x00),$$

где

- K_{in} – ключ диверсификации,
- *label, seed* – значения, фиксируемые и присваиваемые в протоколах.

Функция диверсификации *KDF_GOSTR3411_2012_256* является частным случаем функции *KDF_TREE_GOSTR3411_2012*, описанной в следующем разделе.

5.5 Функция диверсификации KDF_TREE_GOSTR3411_2012_256

Функция диверсификации *KDF_TREE_GOSTR3411_2012_256* на основе $HMAC_{256}$ определяется выражением:

$$KDF\_TREE(K_{in}, label, seed, R) = K(1) \mid K(2) \mid K(3) \mid K(4) \mid \ldots$$

где

$$K(i) = HMAC_{256}\,(K_{in}, [i]_2 \mid label \mid 0x00 \mid seed \mid [L]_2)\ , \quad i \geq 1,$$

где

- R – внешний фиксируемый параметр, с возможными значениями 1, 2, 3, 4,
- K_{in} – ключ диверсификации,
- L – необходимая битовая длина вырабатываемого ключевого материала (целое число), не превосходящее $256\cdot(2^{8\cdot R} - 1)$,
- $[L]_2$ – байтовое представление L, записывается в сетевом порядке байт,
- i – счетчик числа итераций,
- $[i]_2$ – байтовое представление счетчика числа итераций, количество байт в представлении $[i]_2$ равно значению R (не более 4-х байт), записывается в сетевом порядке байт,
- *label, seed* – значения, фиксируемые и присваиваемые в протоколах.

Функция диверсификации *KDF_TREE_GOSTR3411_2012_256* предназначена для порождения ключевого материала длины L, не превосходящей $256\cdot(2^{8\cdot R}-1)$ бит, и использует общие принципы задания входных параметров и выхода для функций диверсификации, изложенные в **NIST SP 800-108** (см. пункт 5.1). В качестве псевдослучайной функции выбран описанный в пункте 5.1 настоящего документа алгоритм $HMAC_{256}$ с выходом 256 бит.

При R = 1 и L = 256 функция *KDF_TREE_GOSTR3411_2012_256* совпадает с функцией *KDF_GOSTR3411_2012_256* из предыдущего раздела.

Каждый ключ, последовательно полученный из ключевого материала, сформированного с помощью ключа диверсификации K_{in} – ключа 0-го уровня, может затем рассматриваться как ключ диверсификации 1-го уровня и также использоваться для генерации ключевого материала. Ключевой материал, полученный из ключа диверсификации 1-го уровня, может быть разбит на ключи диверсификации 2-го уровня. Применение данной процедуры приводит к построению ключевого дерева с корневым ключом K_{in} и формированию ключевого материала с иерархией по уровням, как описано в **NIST SP 800-108** (см. пункт 6). Процедура разбиения ключевого материала на каждом уровне определяется в протоколах.

5.6 Экспорт и импорт ключей

При экспорте секретного ключа K (ключ алгоритма ГОСТ Р 34.10-2012 или алгоритма ГОСТ 28147-89) с использованием заданного ключа экспорта K_e (ключ алгоритма ГОСТ 28147-89) и случайного набора *UKM* длины от 8 до 16 байт формируется экспортное представление ключа K.

В качестве функции диверсификации используется функция *KDF* (см. предыдущий раздел) при фиксированном значении

$$label = (0x26 \mid 0xBD \mid 0xB8 \mid 0x78)$$

и значении *seed*, равном значению *UKM*.

1) Порождается случайный набор *UKM*.
2) С помощью функции диверсификации, использующей в качестве ключа диверсификации ключ экспорта K_e, и в качестве значения seed случайный набор *UKM*, производится формирование ключа, обозначаемого $KEK_e(UKM)$.

$$KEK_e(UKM) = KDF(K_e,label,UKM).$$

3) Вычисляется значение имитовставки по ГОСТ 28147-89 длины 4 байта от данных K на ключе $KEK_e(UKM)$, синхропосылка при этом полагается равной первым 8 байтам *UKM*. Полученный набор обозначается через *CEK_MAC*.
4) Ключ *K* зашифровывается по алгоритму ГОСТ 28147-89 в режиме простой замены с использованием ключа $KEK_e(UKM)$. Результат зашифрования обозначается через *CEK_ENC*.
5) Экспортным представлением ключа полагается набор *(UKM | CEK_ENC | CEK_MAC).*

При импорте ключа по экспортному представлению ключа (ключа алгоритма ГОСТ Р 34.10-2012 или алгоритма ГОСТ 28147-89) и ключу экспорта K_e восстанавливается ключ *K*.

1) Из экспортного представления ключа выделяются наборы *UKM, CEK_E*NC и *CEK_MAC*.
2) С помощью функции диверсификации, использующей в качестве ключа диверсификации ключ экспорта K_e, и в качестве значения *seed* случайный набор *UKM*, производится формирование ключа, обозначаемого $KEK_e(UKM)$.

$$KEK_e(UKM) = KDF(K_e,label,UKM) .$$

3) Набор *CEK_ENC* расшифровывается по алгоритму ГОСТ 28147-89 в режиме простой замены с использованием ключа $KEK_e(UKM)$. Ключ K полагается равным результату расшифрования.
4) Вычисляется значение имитовставки по ГОСТ 28147-89 длины 4 байта от данных K на ключе $KEK_e(UKM)$, синхропосылка при этом полагается равной первым 8 байтам *UKM*. Если результат отличен от *CEK_MAC*, возвращается ошибка.

Алгоритмы экспорта и импорта ключей ГОСТ Р 34.10-2012 являются модификациями алгоритмов *CryptoPro Key Wrap* и *CryptoPro Key Unwrap*, описанных в **RFC4357** (см. пункты 6.3 и 6.4).

5.7 Приложение 1: проверочные примеры

1) HMAC_GOSTR3411_2012_256

Ключ K:

```
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
```

Данные T:

```
01 26 bd b8 78 00 af 21 43 41 45 65 63 78 01 00
```

Значение $HMAC_{256}(K,T)$:

```
a1 aa 5f 7d e4 02 d7 b3 d3 23 f2 99 1c 8d 45 34
01 31 37 01 0a 83 75 4f d0 af 6d 7c d4 92 2e d9
```

2) HMAC_GOSTR3411_2012_512

Ключ K:

```
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
```

Данные T:

```
01 26 bd b8 78 00 af 21 43 41 45 65 63 78 01 00
```

Значение HMAC512(K,T):

```
a5 9b ab 22 ec ae 19 c6 5f bd e6 e5 f4 e9 f5 d8
54 9d 31 f0 37 f9 df 9b 90 55 00 e1 71 92 3a 77
3d 5f 15 30 f2 ed 7e 96 4c b2 ee dc 29 e9 ad 2f
3a fe 93 b2 81 4f 79 f5 00 0f fc 03 66 c2 51 e6
```

3) PRF_TLS_GOSTR3411_2012_256

Ключ K:

```
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
```

Данные seed:

```
18 47 1d 62 2d c6 55 c4 d2 d2 26 96 91 ca 4a 56
0b 50 ab a6 63 55 3a f2 41 f1 ad a8 82 c9 f2 9a
```

Данные label:

```
11 22 33 44 55
```

Выход *T1*:

```
ff 09 66 4a 44 74 58 65 94 4f 83 9e bb 48 96 5f
15 44 ff 1c c8 e8 f1 6f 24 7e e5 f8 a9 eb e9 7f
```

Выход *T2*:

```
c4 e3 c7 90 0e 46 ca d3 db 6a 01 64 30 63 04 0e
c6 7f c0 fd 5c d9 f9 04 65 23 52 37 bd ff 2c 02
```

4) PRF_TLS_GOSTR3411_2012_512

Ключ K:

```
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
```

Данные seed:

```
18 47 1d 62 2d c6 55 c4 d2 d2 26 96 91 ca 4a 56
0b 50 ab a6 63 55 3a f2 41 f1 ad a8 82 c9 f2 9a
```

Данные label:

```
11 22 33 44 55
```

Выход T1:

```
f3 51 87 a3 dc 96 55 11 3a 0e 84 d0 6f d7 52 6c
5f c1 fb de c1 a0 e4 67 3d d6 d7 9d 0b 92 0e 65
ad 1b c4 7b b0 83 b3 85 1c b7 cd 8e 7e 6a 91 1a
62 6c f0 2b 29 e9 e4 a5 8e d7 66 a4 49 a7 29 6d
```

Выход T2:

```
e6 1a 7a 26 c4 d1 ca ee cf d8 0c ca 65 c7 1f 0f
88 c1 f8 22 c0 e8 c0 ad 94 9d 03 fe e1 39 57 9f
72 ba 0c 3d 32 c5 f9 54 f1 cc cd 54 08 1f c7 44
02 78 cb a1 fe 7b 7a 17 a9 86 fd ff 5b d1 5d 1f
```

5) PRF_IPSEC_KEYMAT_GOSTR3411_2012_256

Ключ *K*:

```
c9 a9 a7 73 20 e2 cc 55 9e d7 2d ce 6f 47 e2 19
2c ce a9 5f a6 48 67 05 82 c0 54 c0 ef 36 c2 21
```

Данные *S*:

```
01 26 bd b8 78 00 1d 80 60 3c 85 44 c7 27 01 00
```

Выход *T1*:

```
21 01 d8 0c 47 db 54 bc 3c 82 9b 8c 30 7c 47 55
50 88 83 a6 d6 9e 60 1b f7 aa fb 0a bc a4 ed 95
```

Выход *T2*:

```
33 b8 4e d0 8f 93 56 f8 1d f8 d2 79 f0 79 c9 02
87 cb 45 2c 81 d4 1e 80 38 43 08 86 c1 92 12 aa
```

6) PRF_IPSEC_PRFPLUS_GOSTR3411_2012_256

Ключ K:

```
c9 a9 a7 73 20 e2 cc 55 9e d7 2d ce 6f 47 e2 19
2c ce a9 5f a6 48 67 05 82 c0 54 c0 ef 36 c2 21
```

Данные S:

```
01 26 bd b8 78 00 1d 80 60 3c 85 44 c7 27 01 00
```

Выход T1:

```
2d e5 ee 84 e1 3d 7b e5 36 16 67 39 13 37 0a b0
54 c0 74 b7 9b 69 a8 a8 46 82 a9 f0 4f ec d5 87
```

Выход T2:

```
29 f6 0d da 45 7b f2 19 aa 2e f9 5d 7a 59 be 95
4d e0 08 f4 a5 0d 50 4d bd b6 90 be 68 06 01 53
```

7) PRF_IPSEC_KEYMAT_GOSTR3411_2012_512

Ключ K:

```
c9 a9 a7 73 20 e2 cc 55 9e d7 2d ce 6f 47 e2 19
2c ce a9 5f a6 48 67 05 82 c0 54 c0 ef 36 c2 21
```

Данные S:

```
      01 26 bd b8 78 00 1d 80 60 3c 85 44 c7 27 01 00
```

Выход T1:

```
      b9 55 5b 29 91 75 4b 37 9d a6 8e 60 98 f5 b6 0e
      df 91 8a 56 20 4b ff f3 a8 37 6d 1f 57 ed b2 34
      a5 12 32 81 23 cd 6c 03 0b 54 14 2e 1e c7 78 2b
      03 00 be a5 7c c2 a1 4c a3 b4 f0 85 a4 5c d6 ca
```

Выход T2:

```
      37 b1 e0 86 52 43 a4 fb 29 14 8d 27 4d 30 63 fc
      bf b0 f2 f4 68 d5 27 e4 3b ca 41 fa 6b b5 3e c8
      df 21 bf c4 62 3a 2e 76 8b 64 54 03 3e 09 52 32
      d1 8c 86 a6 8f 00 98 d3 31 81 75 f6 59 05 ae db
```

8) PRF_IPSEC_ PRFPLUS_GOSTR3411_2012_512

Ключ K:

```
      c9 a9 a7 73 20 e2 cc 55 9e d7 2d ce 6f 47 e2 19
      2c ce a9 5f a6 48 67 05 82 c0 54 c0 ef 36 c2 21
```

Данные S:

```
      01 26 bd b8 78 00 1d 80 60 3c 85 44 c7 27 01 00
```

Выход T1:

```
      5d a6 71 43 a5 f1 2a 6d 6e 47 42 59 6f 39 24 3f
      cc 61 57 45 91 5b 32 59 10 06 ff 78 a2 08 63 d5
      f8 8e 4a fc 17 fb be 70 b9 50 95 73 db 00 5e 96
      26 36 98 46 cb 86 19 99 71 6c 16 5d d0 6a 15 85
```

Выход T2:

```
      48 34 49 5a 43 74 6c b5 3f 0a ba 3b c4 6e bc f8
      77 3c a6 4a d3 43 c1 22 ee 2a 57 75 57 03 81 57
      ee 9c 38 8d 96 ef 71 d5 8b e5 c1 ef a1 af a9 5e
      be 83 e3 9d 00 e1 9a 5d 03 dc d6 0a 01 bc a8 e3
```

9) VKO_GOSTR3410_2012_256
с выходом 256 на ключах ГОСТ Р 34.10-2012, 512 бит,
на параметрах id-tc26-gost-3410-12-512-paramSetA

Величина *UKM*:

```
      1d 80 60 3c 85 44 c7 27
```

Закрытый ключ *x* стороны A:

```
      c9 90 ec d9 72 fc e8 4e c4 db 02 27 78 f5 0f ca
      c7 26 f4 67 08 38 4b 8d 45 83 04 96 2d 71 47 f8
      c2 db 41 ce f2 2c 90 b1 02 f2 96 84 04 f9 b9 be
      6d 47 c7 96 92 d8 18 26 b3 2b 8d ac a4 3c b6 67
```

Открытый ключ *x·P* стороны A (точка кривой (X, Y)):

```
      aa b0 ed a4 ab ff 21 20 8d 18 79 9f b9 a8 55 66
      54 ba 78 30 70 eb a1 0c b9 ab b2 53 ec 56 dc f5
      d3 cc ba 61 92 e4 64 e6 e5 bc b6 de a1 37 79 2f
      24 31 f6 c8 97 eb 1b 3c 0c c1 43 27 b1 ad c0 a7
      91 46 13 a3 07 4e 36 3a ed b2 04 d3 8d 35 63 97
```

```
      1b d8 75 8e 87 8c 9d b1 14 03 72 1b 48 00 2d 38
      46 1f 92 47 2d 40 ea 92 f9 95 8c 0f fa 4c 93 75
      64 01 b9 7f 89 fd be 0b 5e 46 e4 a4 63 1c db 5a

Закрытый ключ y стороны B:

      48 c8 59 f7 b6 f1 15 85 88 7c c0 5e c6 ef 13 90
      cf ea 73 9b 1a 18 c0 d4 66 22 93 ef 63 b7 9e 3b
      80 14 07 0b 44 91 85 90 b4 b9 96 ac fe a4 ed fb
      bb cc cc 8c 06 ed d8 bf 5b da 92 a5 13 92 d0 db

Открытый ключ y·P стороны B (точка кривой (X, Y)):

      19 2f e1 83 b9 71 3a 07 72 53 c7 2c 87 35 de 2e
      a4 2a 3d bc 66 ea 31 78 38 b6 5f a3 25 23 cd 5e
      fc a9 74 ed a7 c8 63 f4 95 4d 11 47 f1 f2 b2 5c
      39 5f ce 1c 12 91 75 e8 76 d1 32 e9 4e d5 a6 51
      04 88 3b 41 4c 9b 59 2e c4 dc 84 82 6f 07 d0 b6
      d9 00 6d da 17 6c e4 8c 39 1e 3f 97 d1 02 e0 3b
      b5 98 bf 13 2a 22 8a 45 f7 20 1a ba 08 fc 52 4a
      2d 77 e4 3a 36 2a b0 22 ad 40 28 f7 5b de 3b 79

Значение KEK_VKO:

      c9 a9 a7 73 20 e2 cc 55 9e d7 2d ce 6f 47 e2 19
      2c ce a9 5f a6 48 67 05 82 c0 54 c0 ef 36 c2 21
```

10) VKO_GOSTR3410_2012_512
    с выходом 512 на ключах ГОСТ Р 34.10-2012, 512 бит,
    на параметрах id-tc26-gost-3410-12-512-paramSetA

```
Величина UKM:

      1d 80 60 3c 85 44 c7 27

Закрытый ключ x стороны A:

      c9 90 ec d9 72 fc e8 4e c4 db 02 27 78 f5 0f ca
      c7 26 f4 67 08 38 4b 8d 45 83 04 96 2d 71 47 f8
      c2 db 41 ce f2 2c 90 b1 02 f2 96 84 04 f9 b9 be
      6d 47 c7 96 92 d8 18 26 b3 2b 8d ac a4 3c b6 67

Открытый ключ xP стороны A (точка кривой (X, Y)):

      aa b0 ed a4 ab ff 21 20 8d 18 79 9f b9 a8 55 66
      54 ba 78 30 70 eb a1 0c b9 ab b2 53 ec 56 dc f5
      d3 cc ba 61 92 e4 64 e6 e5 bc b6 de a1 37 79 2f
      24 31 f6 c8 97 eb 1b 3c 0c c1 43 27 b1 ad c0 a7
      91 46 13 a3 07 4e 36 3a ed b2 04 d3 8d 35 63 97
      1b d8 75 8e 87 8c 9d b1 14 03 72 1b 48 00 2d 38
      46 1f 92 47 2d 40 ea 92 f9 95 8c 0f fa 4c 93 75
      64 01 b9 7f 89 fd be 0b 5e 46 e4 a4 63 1c db 5a

Закрытый ключ y стороны B:

      48 c8 59 f7 b6 f1 15 85 88 7c c0 5e c6 ef 13 90
      cf ea 73 9b 1a 18 c0 d4 66 22 93 ef 63 b7 9e 3b
      80 14 07 0b 44 91 85 90 b4 b9 96 ac fe a4 ed fb
      bb cc cc 8c 06 ed d8 bf 5b da 92 a5 13 92 d0 db

Открытый ключ y·P стороны B (точка кривой (X, Y)):
```

```
      19 2f e1 83 b9 71 3a 07 72 53 c7 2c 87 35 de 2e
      a4 2a 3d bc 66 ea 31 78 38 b6 5f a3 25 23 cd 5e
      fc a9 74 ed a7 c8 63 f4 95 4d 11 47 f1 f2 b2 5c
      39 5f ce 1c 12 91 75 e8 76 d1 32 e9 4e d5 a6 51
      04 88 3b 41 4c 9b 59 2e c4 dc 84 82 6f 07 d0 b6
      d9 00 6d da 17 6c e4 8c 39 1e 3f 97 d1 02 e0 3b
      b5 98 bf 13 2a 22 8a 45 f7 20 1a ba 08 fc 52 4a
      2d 77 e4 3a 36 2a b0 22 ad 40 28 f7 5b de 3b 79
```

Значение KEKVKO:

```
      79 f0 02 a9 69 40 ce 7b de 32 59 a5 2e 01 52 97
      ad aa d8 45 97 a0 d2 05 b5 0e 3e 17 19 f9 7b fa
      7e e1 d2 66 1f a9 97 9a 5a a2 35 b5 58 a7 e6 d9
      f8 8f 98 2d d6 3f c3 5a 8e c0 dd 5e 24 2d 3b df
```

11) Функция диверсификации KDF_GOSTR3411_2012_256:

Ключ K_{in}:

```
      00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
      10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
```

label:

```
      26 bd b8 78
```

seed:

```
      af 21 43 41 45 65 63 78
```

Значение KDF(K_{in}, label, seed):

```
      a1 aa 5f 7d e4 02 d7 b3 d3 23 f2 99 1c 8d 45 34
      01 31 37 01 0a 83 75 4f d0 af 6d 7c d4 92 2e d9
```

12) Функция диверсификации KDF_TREE_GOSTR3411_2012_256

```
Длина выхода L: 512

Ключ Kin:

      00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
      10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f

label:

      26 bd b8 78

  seed:

      af 21 43 41 45 65 63 78

  Значение K1:

      22 b6 83 78 45 c6 be f6 5e a7 16 72 b2 65 83 10
      86 d3 c7 6a eb e6 da e9 1c ad 51 d8 3f 79 d1 6b

  Значение K2:

     07 4c 93 30 59 9d 7f 8d 71 2f ca 54 39 2f 4d dd
     e9 37 51 20 6b 35 84 c8 f4 3f 9e 6d c5 15 31 f9
```

13) Экспорт и импорт ключей
на параметрах szOID_Gost28147_89_TC26_Z_ParamSet

Ключ K:

```
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
```

Величина UKM:

```
af 21 43 41 45 65 63 78
```

label:

```
26 bd b8 78
```

$KEK_e(UKM) = KDF(K_e,label,UKM)$:

```
a1 aa 5f 7d e4 02 d7 b3 d3 23 f2 99 1c 8d 45 34
01 31 37 01 0a 83 75 4f d0 af 6d 7c d4 92 2e d9
```

CEK_MAC:

```
38 d5 8a a3
```

CEK_ENC:

```
b9 fb 92 42 95 0f 84 3f 0f bd 5b 9a 5e cf 9f 17
f7 9e 6d 21 58 16 56 de 6d c5 85 dd 62 7a 44 0a
```