
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ
(РОССТАНДАРТ)

Технический комитет 026

«Криптографическая защита информации»

Информационная технология

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

РЕКОМЕНДАЦИИ ПО СТАНДАРТИЗАЦИИ

**ИСПОЛЬЗОВАНИЕ НАБОРОВ АЛГОРИТМОВ
ШИФРОВАНИЯ НА ОСНОВЕ ГОСТ 28147-89
ДЛЯ ПРОТОКОЛА БЕЗОПАСНОСТИ
ТРАНСПОРТНОГО УРОВНЯ
(TLS)**

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

Москва
2014

Введение

Настоящие рекомендации предназначены для регистрации новых наборов алгоритмов шифрования протокола безопасности транспортного уровня (Transport Layer Security, TLS) в соответствии с процедурой, указанной в стандартах протокола TLS. Эти наборы алгоритмов шифрования основаны на российских государственных криптографических стандартах – открытые ключи по ГОСТ Р 34.10, алгоритм шифрования по ГОСТ 28147-89, алгоритмы хэширования по ГОСТ Р 34.11.

Необходимость разработки настоящего документа вызвана потребностью в обеспечении совместимости реализаций протокола TLS с использованием российских государственных криптографических стандартов.

С о д е р ж а н и е

1	Область применения	4
2	Нормативные ссылки	4
2.1	Дополнительные ссылки	5
3	Обозначения и сокращения	5
4	Определения наборов алгоритмов шифрования.....	6
4.1	Сертификаты сторон	6
4.2	Обмен ключами.....	6
4.3	Псевдослучайная функция	6
4.4	Алгоритм шифрования и MAC	7
4.5	Алгоритм подписи и хэширования	8
5	Структуры данных и вычисления.....	8
5.1	Алгоритм шифрования	8
5.2	Вычисление ключей.....	8
5.3	Сертификат сервера (Server Certificate)	8
5.4	Сообщение об обмене ключа со стороны сервера (Server Key Exchange Message) ...	8
5.5	Запрос на получение сертификата (Certificate Request)	8
5.6	Сообщение об обмене ключа со стороны клиента (Client Key Exchange Message)	9
5.7	Проверка сертификата (Certificate Verify)	10
5.8	Завершение (Finished).....	10
6	Совместимость.....	10
7	Вопросы безопасности	12
8	Согласование с IANA	12
8.1	Приватные типы алгоритмов	13
Приложение А	Примеры (информативное)	14
A.1.	TLS 1.0 аутентификация клиента с ключом 512 бит и сервера с ключом 512 бит	14
A.2.	TLS 1.0 аутентификация клиента с ключом 256 бит и сервера с ключом 256 бит	24
A.3.	TLS 1.0 аутентификация клиента с ключом 256 бит и сервера с ключом 512 бит	35
A.4.	TLS 1.0 аутентификация сервера с ключом 256 бит	46
Приложение В	Модули ASN.1	56
Библиография	58

1 Область применения

Настоящие методические рекомендации предназначены для регистрации новых наборов алгоритмов шифрования протокола безопасности транспортного уровня (Transport Layer Security, TLS) в соответствии с процедурой, указанной в стандартах протокола TLS. Эти наборы алгоритмов шифрования основаны на российских государственных криптографических стандартах – открытые ключи по ГОСТ Р 34.10, алгоритм шифрования по ГОСТ 28147-89, алгоритм хэширования по ГОСТ Р 34.11.

В этом документе определяются две конфигурации: анонимный клиент – аутентифицированный сервер (сертификат предоставляет только сервер); аутентифицированный клиент - аутентифицированный сервер (клиент и сервер обмениваются сертификатами).

Используемый здесь язык представления такой же, как и в [IETF RFC 5246].

Поскольку настоящая спецификация расширяет [IETF RFC 5246], её содержание должно быть объединено с содержанием спецификации протокола TLS и другими спецификациями, которые расширяют протокол TLS.

2 Нормативные ссылки

В настоящем документе использованы нормативные ссылки на следующие стандарты и рекомендации:

ГОСТ 28147 - «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования», ГОСТ 28147-89, Государственный стандарт Союза ССР, Государственный комитет СССР по стандартам, ИПК Издательство стандартов, 1996.

ГОСТ Р 34.10 - «Информационные технологии. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи», ГОСТ Р 34.10-2012, Национальный стандарт Российской Федерации, Федеральное агентство по техническому регулированию и метрологии, Стандартинформ, 2012.

ГОСТ Р 34.11 - «Информационные технологии. Криптографическая защита информации. Функция хэширования», ГОСТ Р 34.11-2012, Национальный стандарт Российской Федерации, Федеральное агентство по техническому регулированию и метрологии, Стандартинформ, 2012.

ГОСТ Р ИСО/МЭК 8824-1 - «Информационные технологии. Абстрактная синтаксическая нотация версии один (ASN.1). Часть 1. Спецификация основной нотации», ГОСТ Р ИСО/МЭК 8824-1-2001, Государственный стандарт Российской Федерации, Госстандарт России, Москва, 2001.

ГОСТ Р ИСО/МЭК 8825-1 - «Информационные технологии. Правила кодирования ASN.1. Часть 1. Спецификация базовых (BER), канонических (CER) и отличительных (DER) правил кодирования», ГОСТ Р ИСО/МЭК 8825-1-2003, Государственный стандарт Российской Федерации, Госстандарт России, Москва, 2003.

TK26АЛГ — Федеральное агентство по техническому регулированию и метрологии (РОССТАНДАРТ), Технический комитет №26, «Рекомендации по стандартизации. Использование криптографических алгоритмов, сопутствующих применению стандартов ГОСТ Р 34.10-2012 и ГОСТ Р 34.11-2012», Москва, 2014.

TK26ИОК — Федеральное агентство по техническому регулированию и метрологии (РОССТАНДАРТ), Технический комитет №26, «Техническая спецификация использования алгоритмов ГОСТ Р 34.10, ГОСТ Р 34.11 в профиле сертификата и списке отзыва сертификатов (CRL) инфраструктуры открытых ключей X.509», Москва, 2014.

TK26СMS — Федеральное агентство по техническому регулированию и метрологии (РОССТАНДАРТ), Технический комитет №26, «Рекомендации по стандартизации. Использование алгоритмов ГОСТ 28147-89, ГОСТ Р 34.11 и ГОСТ Р 34.10 в криптографических сообщениях формата CMS», Москва, 2014.

TK26УЗ — Федеральное агентство по техническому регулированию и метрологии (РОССТАНДАРТ), Технический комитет №26, «Рекомендации по стандартизации. Задание узлов замены блока подстановки алгоритма шифрования ГОСТ 28147-89», Москва, 2013.

2.1 Дополнительные ссылки

IETF RFC 2246 - Т. Дьеркс и К. Аллен, «Протокол TLS версии 1.0» (Dierks, T. and C. Allen, The TLS Protocol Version 1.0), RFC 2246, январь 1999.

IETF RFC 5246 - Т. Дьеркс и Э. Рескола, «Протокол протокола безопасности транспортного уровня (TLS) версии 1.2» (Dierks, T. and E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.2), RFC 5246, июнь 2006.

Примечание – При пользовании настоящим документом целесообразно проверить действие ссылочных стандартов и классификаторов в информационной системе общего пользования - на официальном сайте национального органа Российской Федерации по стандартизации в сети Интернет или по ежегодно издаваемому информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по соответствующим ежемесячно издаваемым информационным указателям, опубликованным в текущем году. Если ссылочный документ заменён (изменён), то при пользовании настоящим документом следует руководствоваться заменённым (изменённым) документом. Если ссылочный документ отменен без замены, то положение, в котором дана ссылка на него, применяется в части, не затрагивающей эту ссылку.

3 Обозначения и сокращения

В настоящих рекомендациях используются следующие обозначения и сокращения:

Key Meshing	Алгоритм усложнения ключа [TK26CMS].
CipherSuite	Набор алгоритмов шифрования, обеспечения целостности и аутентичности в протоколах семейства SSL/TLS.
EAP	Расширяемый протокол аутентификации (Extensible Authentication Protocol).
HASH_GOSTR3411_SUITE	Функция хэширования по ГОСТ Р 34.11, конкретный алгоритм и параметры определяются CipherSuite (HASH_GOSTR341194, HASH_GOSTR341112_256).
HASH_GOSTR3411_CERT	Функция хэширования по ГОСТ Р 34.11, конкретный алгоритм и параметры определяются сертификатом клиента (HASH_GOSTR341194, HASH_GOSTR341112_256, HASH_GOSTR341112_512).
HMAC	Код аутентификации сообщения на основе хэш-значения (Hash-based Message Authentication Code).
HMAC_GOSTR3411	Код аутентификации сообщения на основе хэш-кода, вычисленного согласно ГОСТ Р 34.11, конкретный алгоритм и параметры определяется CipherSuite (HMAC_GOSTR341194, HMAC_GOSTR3411_2012_256). Функция на основе ГОСТ Р 34.11-94 определена в [IETF RFC 4357], функции на основе ГОСТ Р 34.11-2012 определены в [TK26АЛГ].
HMAC_GOSTR341194	Функция на основе ГОСТ Р 34.11-94, определена в [IETF RFC 4357] как HMAC_GOSTR3411.
IMIT_GOST28147	Алгоритм по ГОСТ 28147 в режиме имитовставки.

MAC	Код аутентификации сообщения (Message Authentication Code).
PRF	Семейство псевдослучайных функций (Pseudorandom Function family).
PRF_GOSTR3411	Псевдослучайная функция на основе кода аутентификации HMAC_GOSTR3411. Функция на основе ГОСТ Р 34.11-94 определена в [IETF RFC 4357], функции на основе ГОСТ Р 34.11-2012 определены в [TK26АЛГ].
PRF_GOSTR341194	Псевдослучайная функция на основе кода аутентификации HMAC_GOSTR341194, определена в [IETF RFC 4357] как PRF_GOSTR3411.
UKM	Материал ключа пользователя (user key material).

4 Определения наборов алгоритмов шифрования

Настоящие методические рекомендации определяют четыре поточных набора алгоритмов шифрования (CipherSuite, Stream ciphers):

- TLS_GOSTR341112_256_WITH_28147_CNT_IMIT;
- TLS_GOSTR341112_256_WITH_NULL_GOSTR3411;
- TLS_GOSTR341001_WITH_28147_CNT_IMIT;
- TLS_GOSTR341001_WITH_NULL_GOSTR3411.

Предназначенные для использования в протоколе TLS версии 1.2.

4.1 Сертификаты сторон

Клиент TLS, поддерживающий настоящие рекомендации, может поддерживать все виды хэш функций, подписей и все типы сертификатов для всех CipherSuite, описанных в данном документе.

Сервер TLS имеющий сертификат ключа по ГОСТ Р 34.10-2012 для любой длины ключа может поддерживать CipherSuite:

- TLS_GOSTR341112_256_WITH_28147_CNT_IMIT;
- TLS_GOSTR341112_256_WITH_NULL_GOSTR3411.

Сервер TLS имеющий сертификат ключа по ГОСТ Р 34.10-2001 может поддерживать CipherSuite:

- TLS_GOSTR341001_WITH_28147_CNT_IMIT;
- TLS_GOSTR341001_WITH_NULL_GOSTR3411;
- TLS_GOSTR341112_256_WITH_28147_CNT_IMIT;
- TLS_GOSTR341112_256_WITH_NULL_GOSTR3411.

4.2 Обмен ключами

Определённые здесь наборы алгоритмов шифрования для передачи premaster_secret (256-бит) используют алгоритм передачи ключей определённый сертификатом сервера [TK26CMS].

4.3 Псевдослучайная функция

Описанные здесь наборы алгоритмов шифрования используют HMAC и псевдослучайную функцию (PRF) TLS в соответствии с разделом 5 [IETF RFC 5246], на основе функции хэширования ГОСТ Р 34.11 (HASH_GOSTR3411_SUITE, HMAC_GOSTR3411, PRF_GOSTR3411). Алгоритм ГОСТ Р 34.11 и его параметры определяются согласованным набором алгоритмов шифрования

Т а б л и ц а 1. Алгоритмы хэширования и псевдослучайной функции

Набор алгоритмов шифрования	Хэш, HMAC и псевдослучайная функция
TLS_GOSTR341112_256_WITH_28147_CNT_IMIT	HASH_GOSTR341112_256 HMAC_GOSTR3411_2012_256 PRF_TLS_GOSTR3411_2012_256
TLS_GOSTR341001_WITH_28147_CNT_IMIT	HASH_GOSTR341194 HMAC_GOSTR341194 PRF_GOSTR341194
TLS_GOSTR341112_256_WITH_NULL_GOSTR3411	HASH_GOSTR341112_256 HMAC_GOSTR3411_2012_256 PRF_TLS_GOSTR3411_2012_256
TLS_GOSTR341001_WITH_NULL_GOSTR3411	HASH_GOSTR341194 HMAC_GOSTR341194 PRF_GOSTR341194

Зависимые протоколы, такие как SSTP, EAP-TLS [IETF RFC 5216], ДОЛЖНЫ использовать согласованную PRF.

4.4 Алгоритм шифрования и MAC

Используются следующие алгоритмы шифрования и MAC-функции (подробное описание алгоритмов шифрования приведено в разделе 5.1 настоящего документа):

Таблица 2. Алгоритмы шифрования и MAC-функции

Набор алгоритмов шифрования	Алгоритм шифрования и параметры	MAC
TLS_GOSTR341112_256_WITH_28147_CNT_IMIT	ГОСТ 28147-89 id-tc26-gost-28147-param-Z	IMIT_GOST28147
TLS_GOSTR341001_WITH_28147_CNT_IMIT	ГОСТ 28147-89 id-Gost28147-89-CryptoPro-A-ParamSet	IMIT_GOST28147
TLS_GOSTR341112_256_WITH_NULL_GOSTR3411	–	HMAC_GOSTR3411_2012_256
TLS_GOSTR341001_WITH_NULL_GOSTR3411	–	HMAC_GOSTR341194

Параметры алгоритма шифрования ГОСТ 28147-89 и IMIT_GOST28147 определяются согласованным набором алгоритмов шифрования.

Для стандартных потоковых CipherSuite MAC рассчитывается на следующие данные [IETF RFC 5246]:

```
MACed_data[seq_num] = seq_num +
                        TLSCompressed.type +
                        TLSCompressed.version +
                        TLSCompressed.length +
                        TLSCompressed.fragment;
```

Определённые в настоящих рекомендациях CipherSuite используют эти входные данные конкатенированные со всеми предыдущими записями. Таким образом, входными данными функций IMIT_GOST28147 и HMAC_GOSTR3411 являются:

```
MAC (MACed_data[n]) = IMIT_GOST28147 (MACed_data[0] + ... + MACed_data[n]);  
MAC (MACed_data[n]) = HMAC_GOSTR3411 (MACed_data[0] + ... + MACed_data[n])
```

4.5 Алгоритм подписи и хэширования

Алгоритм и параметры подписи ГОСТ Р 34.10 и функции хэширования, которые используются при вычислении и проверке CertificateVerify, определяются выбором сертификата клиента.

Примечание:

С учетом того, что набор входных данных функции хэширования является конкатенацией всех сообщений вплоть до сообщения CertificateVerify (п. 7.4.8 [IETF RFC 5246]), совместимые реализации могут как накапливать эти сообщения, так и рассчитывать все потенциально возможные функции хэширования (функции хэширования для всех потенциально возможных сертификатов, [IETF RFC 2246]).

5 Структуры данных и вычисления

5.1 Алгоритм шифрования

ГОСТ 28147-89 использует 256-битный ключ и 8-байтовый вектор инициализации (IV).

Определённые здесь наборы алгоритмов шифрования используют ГОСТ 28147-89 в качестве потокового алгоритма шифрования в режиме гаммирования, параметры определяются согласованным набором алгоритмов шифрования.

IMIT_GOST28147 – это алгоритм по ГОСТ 28147-89 в режиме имитовставки (4 байта).

5.2 Вычисление ключей

Вычисление ключей производится в соответствии с разделом 6.3 [IETF RFC 5246] с использованием функции PRF_GOSTR3411. Используются следующие параметры:

- SecurityParameters.enc_key_length = 32
- SecurityParameters.mac_key_length = 32
- SecurityParameters.fixed_iv_length = 8

Длина необходимого ключевого материала составляет 144 байта.

5.3 Сертификат сервера (Server Certificate)

Для данных наборов алгоритмов шифрования требуется такое сообщение, и оно ДОЛЖНО содержать сертификат с алгоритмом открытого ключа, соответствующим полю ServerHello.cipher_suite.

5.4 Сообщение об обмене ключа со стороны сервера (Server Key Exchange Message)

В этих наборах алгоритмов шифрования данное сообщение использоваться НЕ ДОЛЖНО, так как все необходимые параметры присутствуют в сертификате сервера [ТК26ИОК].

5.5 Запрос на получение сертификата (Certificate Request)

Это сообщение используется в соответствии с [IETF RFC 5246] и расширяется следующим образом:

```
enum {  
    gostr34102001(22), gostr34102012_256(<TBD>+2),  
    gostr34102012_512(<TBD>+3), (255)  
} ClientCertificateType;
```

Типы сертификатов gostr34102001, gostr34102012_256, gostr34102012_512 указывают на то, что сервер принимает сертификаты с открытыми ключами по ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012 с соответствующим размером закрытого ключа.


```

enum{
    gostr3411(<TBD>+1), gostr34112012_256(<TBD>+2),
    gostr34112012_512(<TBD>+3), (255)
} HashAlgorithm;
enum{
    gostr34102001(<TBD>+1), gostr34102012_256(<TBD>+3),
    gostr34102012_512(<TBD>+3), (255)
} SignatureAlgorithm;

```

Пары SignatureAndHashAlgorithm, (gostr3411, gostr34102001), (gostr34112012_256, gostr34102012_256) и (gostr34112012_512, gostr34102012_512) указывают на то, что сервер поддерживает данные алгоритмы при обработке сертификатов.

5.6 Сообщение об обмене ключа со стороны клиента (Client Key Exchange Message)

Это сообщение используется в соответствии с [IETF RFC 5246]. Оно необходимо для данных наборов алгоритмов шифрования и содержит закодированную в DER структуру TLSGostKeyTransportBlob [ГОСТ Р ИСО/МЭК 8825-1].

```

enum { vko_gost } KeyExchangeAlgorithm;

struct {
    select (KeyExchangeAlgorithm) {
        case vko_gost: TLSGostKeyTransportBlob;
    } exchange_keys;
} ClientKeyExchange;

```

Определение ASN.1 для этой структуры выглядит следующим образом:

```

TLSGostKeyTransportBlob ::= SEQUENCE {
    keyBlob          GostR3410-KeyTransport,
    proxyKeyBlobs   SEQUENCE OF TLSProxyKeyTransportBlob OPTIONAL
}

```

Определение структуры GostR3410-KeyTransport смотри в [TK26CMS], в структуре keyBlob ДОЛЖНО присутствовать поле transportParameters.

В том случае, если сервер не запрашивал сертификат клиента или алгоритм и параметры открытого ключа клиента не совпадают с алгоритмом и параметрами открытого ключа сервера, то ДОЛЖНО присутствовать поле keyBlob.transportParameters.ephemeralPublicKey. В противном случае данное поле СЛЕДУЕТ пропустить.

```

TLSProxyKeyTransportBlob ::= SEQUENCE {
    keyBlob          GostR3410-KeyTransport,
    cert             OCTET STRING
}

```

прохуKeyBlobs – (не обязательно) содержит сообщения для обмена ключами для вторичных получателей (например, для межсетевого экрана, который проверяет соединения).

cert - содержит сертификат вторичного получателя.

5.6.1 Действия клиента

Клиент формирует структуру GostR3410-KeyTransport для передачи одноразового (случайного) 256-битного ключа premaster_secret согласно [TK26CMS], для вычисления UKM используется хэш-функция HASH_GOSTR3411_SUITE определяемая согласованным набором алгоритмов шифрования:

```

UKM = HASH_GOSTR3411_SUITE(client_random|server_random) [0..7]

```

Если сервер запрашивает сертификат клиента, а так же если алгоритм и параметры сертификатов сервера и клиента совпадают, то в качестве ключа отправителя ДОЛЖЕН быть

использован секретный ключ, соответствующий сертификату клиента. В противном случае клиент заполняет поле `keyBlob.transportParameters.ephemeralPublicKey`, и в качестве ключа отправителя ДОЛЖЕН быть использован соответствующий секретный ключ.

5.6.2 Действия сервера

Сервер обрабатывает структуру `GostR3410-KeyTransport` согласно [TK26CMS], предварительно сервер ДОЛЖЕН проверить значение UKM до начала расшифрования `premaster_secret`.

5.7 Проверка сертификата (Certificate Verify)

Если клиент предоставил и сертификат клиента, и эфемерный открытый ключ, он ДОЛЖЕН отправить сообщение о проверке сертификата в качестве доказательства владения закрытым ключом для предоставленного сертификата.

При использовании `CipherSuite` в рамках TLS 1.2 [IETF RFC 5246]. Расширений структур не требуется.

5.8 Завершение (Finished)

Это сообщение используется в соответствии с описанием, приведённым в разделе 7.4.9 из [IETF RFC 5246]. Используются алгоритмы `PRF_GOSTR3411` и `HASH_GOSTR3411_SUITE`, определяемые согласованным набором алгоритмов шифрования:

```
Finished.verify_data = PRF_GOSTR3411(master_secret, finished_label,  
                                HASH_GOSTR3411_SUITE(handshake_messages)) [0..11]
```

6 Совместимость

Требования по реализации TLS на основе ГОСТ 28147-89:

- `TLS_GOSTR341112_256_WITH_28147_CNT_IMIT` – обязательно;
- `gostr34112012_256` и `gostr34102012_256` - обязательно.

По историческим причинам некоторые TLS сервера имеющие только один сертификат ключа по ГОСТ Р 34.10-2001, при отсутствии перечисленных `CipherSuite` в списке `ClientHello.cipher_suites`, могут отвечать `ServerHello.cipher_suite` со значением `TLS_GOSTR341001_WITH_28147_CNT_IMIT`.

По историческим причинам некоторые приложения используют наборы алгоритмов шифрования, определённые в настоящем документе в протоколе TLS версии 1.0 [IETF RFC 2246], частично используя функционал из [IETF RFC 5246], включая вычисление псевдослучайной функции (PRF), зависимой от набора алгоритмов шифрования, сообщения о завершении (Finished) и сообщения о проверке сертификата (Certificate Verify).

При использовании `CipherSuite` в рамках TLS 1.0 [IETF RFC 2246], тип `Signature` должен быть расширен:

```
select (SignatureAlgorithm) {  
  case gostr34102001:  
  case gostr34102012_256:  
    digitally-signed struct {  
      opaque gostr3411_256_hash[32];  
    };  
  case gostr34102012_512:  
    digitally-signed struct {  
      opaque gostr3411_512_hash[64];  
    };  
} Signature;
```

При использовании этого типа в сообщении "Certificate Verify", хэш-код вычисляется с помощью согласованной хэш функции от всех сообщений handshake: HASH_GOSTR3411_CERT(handshake_messages).

7 Вопросы безопасности

Перед началом использования значений подписи, открытых ключей субъекта и параметров алгоритма приложениям РЕКОМЕНДУЕТСЯ проводить проверку на предмет их соответствия стандартам ГОСТ Р 34.10.

И клиенту, и серверу РЕКОМЕНДУЕТСЯ выполнять проверку срока использования закрытого ключа другой стороны.

При согласовании используемой CipherSuite согласно [IETF RFC 5246]: сервер выбирает первую, удовлетворяющую политике безопасности сервера, CipherSuite из упорядоченного списка ClientHello.cipher_suites, который формируется согласно списку предпочтений политики безопасности клиента. При отсутствии дополнительных требований рекомендуется, чтобы TLS_GOSTR341112_256_WITH_28147_CNT_IMIT, TLS_GOSTR341112_256_WITH_NULL_GOSTR3411 предшествовали TLS_GOSTR341001_WITH_28147_CNT_IMIT и TLS_GOSTR341001_WITH_NULL_GOSTR3411 в списке предпочтений политики безопасности клиента.

При согласовании используемого сертификата клиента согласно [IETF RFC 5246]: клиент выбирает первый сертификат (с цепочкой сертификатов) удовлетворяющий политике безопасности клиента из упорядоченного списка поля supported_signature_algorithms (certificate_types) с ограничением по списку УЦ в поле certificate_authorities структуры CertificateRequest, который формируется согласно списку предпочтений политики безопасности сервера. При отсутствии дополнительных требований рекомендуется, чтобы gostr34112012_512, gostr34112012_256 (gostr34102012_512, gostr34102012_256) предшествовали gostr3411 (gostr34102001) в списке предпочтений политики безопасности сервера.

8 Согласование с IANA

В этом документе определяются следующие новые наборы алгоритмов шифрования, значения которых используются в нескольких реализациях наборов алгоритмов шифрования протокола TLS 1.0. В настоящее время они перечислены в реестре в качестве зарезервированных. Произведён запрос в IANA на обновление реестра набора алгоритмов шифрования протокола TLS, определённого в [IETF RFC 5246] такими значениями:

- CipherSuite TLS_GOSTR341001_WITH_28147_CNT_IMIT = {0x00,0x81}
- CipherSuite TLS_GOSTR341001_WITH_NULL_GOSTR3411 = {0x00,0x83}

В этом документе определяются следующие новые типы клиентских сертификатов. Их значения, представленные здесь, используются несколькими реализациями одних и тех же наборов алгоритмов шифрования протокола TLS 1.0 и описаны в предыдущих проектах.

В настоящее время они перечислены в реестре в качестве зарезервированных. Произведён запрос в IANA на обновление реестра идентификаторов TLS ClientCertificateType, определённого в [IETF RFC 5246], следующими значениями:

```
enum {
    gostr34102001(22), gostr34102012_256(<TBD>+2),
    gostr34102012_512(<TBD>+3), (255)
} ClientCertificateType;
```

В этом документе определяются следующие новые типы алгоритмов подписи, чьи значения следует присвоить из реестра TLS SignatureAlgorithm, определённого в [IETF RFC 5246]:

```
enum{
    gostr34102001(<TBD>+1), gostr34102012_256(<TBD>+3),
    gostr34102012_512(<TBD>+3), (255)
} SignatureAlgorithm;
```

В этом документе определяются следующие новые типы алгоритма хэширования, чьи значения следует присвоить из реестра TLS HashAlgorithm, определённого в [IETF RFC 5246]:

```
enum{
    gostr3411(<TBD>+1), gostr34112012_256(<TBD>+2),
    gostr34112012_512(<TBD>+3), (255)
} HashAlgorithm;
```

8.1 Приватные типы алгоритмов

До регистрации в IANA предварительные реализации используют следующие приватные номера преобразований:

- CipherSuite TLS_GOSTR341112_256_WITH_28147_CNT_IMIT = {0xFF,0x85}
- CipherSuite TLS_GOSTR341112_256_WITH_NULL_GOSTR3411 = {0xFF,0x87}

```
enum {
    gostr34102001(22), gostr34102012_256(238),
    gostr34102012_512(239), (255)
} ClientCertificateType;
enum{
    gostr34102001(237), gostr34102012_256(238),
    gostr34102012_512(239), (255)
} SignatureAlgorithm;
enum{
    gostr3411(237), gostr34112012_256(238),
    gostr34112012_512(239), (255)
} HashAlgorithm;
```

Приложение А Примеры (информативное)

А.1. TLS 1.0 аутентификация клиента с ключом 512 бит и сервера с ключом 512 бит

client -> server:

```
header type
16 (handshake)
major version
03
minor version
01
length
004F
message type
01 (client hello)
Length
00004B
TLS VERSION
                                major version
                                03
                                minor version
                                01
client random
52E78ECF 8CF81884 8F016AA5 2ED29277 66015AC6 24648148 A58D736D 83216A7D
session id length
00
session id

ciphersuite length
0004
ciphersuite
0081          TLS_CIPHER_2001
FF85          TLS_CIPHER_2012
compression methods
                                length
                                01
                                compression method
                                00

extensions
  extensions length
  001E
  renegotiation info (FF01)
  length
  0001
  Renegotiation Info extension
  00
  server name (0000)
  length
  0015
  server name list length
  0013
  server name type
```


102311F1 3F585D50 D6456DA3 7B3E6E63 9D0F584A 6AB114F7 68E27619 92A38201
56308201 52301306 03551D25 040C300A 06082B06 01050507 0302300B 0603551D
0F040403 02043030 1D060355 1D0E0416 0414E8EF 647D662D 0B86647A D9525E02
A8ABF409 6A43301F 0603551D 23041830 1680149E 03F0B89C FC60DC8A 181EE800
DFA85B32 CD737630 3F060355 1D1F0438 30363034 A032A030 862E6874 74703A2F
2F766D2D 74657374 2D63612E 63702E72 752F4365 7274456E 726F6C6C 2F746573
742D6361 2E63726C 3081AC06 082B0601 05050701 0104819F 30819C30 4B06082B
06010505 07300286 3F687474 703A2F2F 766D2D74 6573742D 63612E63 702E7275
2F436572 74456E72 6F6C6C2F 766D2D74 6573742D 63612E63 702E7275 5F746573
742D6361 2E637274 304D0608 2B060105 05073002 86416669 6C653A2F 2F5C5C76
6D2D7465 73742D63 612E6370 2E72755C 43657274 456E726F 6C6C5C76 6D2D7465
73742D63 612E6370 2E72755F 74657374 2D63612E 63727430 0806062A 85030202
03034100 DE24ECA8 3C9A6226 4AF6D978 4E71AE0D FB4694CC CE21A16E 3D1EA426
B8E1A4C2 274C762A 0F6A1CE2 087116E1 0DDDB2B5 A6AA3F99 7F4BA6CD D8FCF026
623B5A62

premastersecret

0619ECC1 161CB54D 391B3610 CEDD90E5 C858DC3C AF199005 C45F9821 2B21A761

client private key

D95C536A 47657160 032FAEE5 CE9465DE DACE5058 B0F51891 87A21CA5 87A1C4AA

408AD861 230A239E 7BD15632 EE1DEDFC CBF9A051 F46C6186 02309605 31AEB500

server public key blob header

06200100 3D2E0000 4D414731 00040000 30150609 2A850307 01020102 0106082A

85030701 010203F0

server public key

F0DBBD44 A22D19AF CB22A0CA D421A02E 7930D5C6 E549C13B BF7D1437 7C67CAE8

7D45E79E 27E96D63 1A0AA6C7 1E6B353C 66A02362 F9D43FC3 F53DDFAB 567CDFB9

2909DCB6 2A931896 ED1F1D16 55AC584D C8D6745C 51EA68CA C8EBF49C AC305EFA

64285097 99F30D21 9A840782 7EB27629 3E7BC1AE 68DCB939 FAD9BFBC AD938C84

hash(client_random, server_random)

F2249B00 0ACEDF36 FF45A3F3 F71C5EBC E45723CC C73C86BA 1BF8C287 97CD9539

ukm

F2249B00 0ACEDF36

Inner Point

5DB5F416 EBE212CF F33C790C BCF2E39F 28F1C24A 731CB518 AA0BB418 B9BF4F0F

EB71132F A1677F4B A033340A 5093543A B1C2C273 D25C94D4 6D24D63A ACD5D9D1

9CEFFF01 719A677B 0A6DE8A1 605E243F 267452B8 4BDECB5E CE36A881 2D2E4ACC

A805B1E2 89331E73 2E695FD6 AD5D3C6D 916E6371 721D4D3A B4F8D4D4 325E5354

VKO result

7835B7D8 5E0E3B28 8D99BF0E 3451C5FB 8F63808E EF9B3DEA 2066A000 AEB10358

diversified key

BE797602 638CEFBB BB338DEF 1AA233BF A6B5F254 9F33AAA1 21E30000 A3FE66E2

premastersecret encrypted on exchange key using parameters

1.2.643.7.1.2.5.1.1 (id-tc26-gost-28147-param-Z) in ECB mode

37C36990 7DAA1C4C 195B7C02 2C4CFA09 AE6A7F60 124EE1AB

8443EA35 A6FDE0B0

MAC on exchange key using parameters

1.2.643.7.1.2.5.1.1 (id-tc26-gost-28147-param-Z) with ukm as IV

D250C063

client key exchange

30433041 30280420 37C36990 7DAA1C4C 195B7C02 2C4CFA09 AE6A7F60 124EE1AB

8443EA35 A6FDE0B0 0404D250 C063A015 06092A85 03070102 05010104 08F2249B

000ACEDF 36


```
A7A6A5A4 ABAAA9A8 AFAEADAC B3B2B1B0 B7B6B5B4 BBBAB9B8 BFBEBDBC C3C2C1C0
C7C6C5C4 CBCAC9C8 CFCECDCC D3D2D1D0 D7D6D5D4 DBDAD9D8 DFDEDDDC E3E2E1E0
E7E6E5E4 EBBAE9E8 EEEEEDEC F3F2F1F0 F7F6F5F4 FBFAF9F8 020100FC 06050403
0A090807 0E0D0C0B 1211100F 16151413 1A191817 1E1D1C1B 2221201F 26252423
2A292827 2E2D2C2B 3231302F 36353433 3A393837 3E3D3C3B 4241403F 46454443
4A494847 4E4D4C4B 5251504F 56555453 5A595857 5E5D5C5B 6261605F 66656463
6A696867 6E6D6C6B 7271706F 76757473 7A797877 7E7D7C7B 8281807F 86858483
8A898887 8E8D8C8B 9291908F 96959493 9A999897 9E9D9C9B A2A1A09F A6A5A4A3
AAA9A8A7 AEADACAB B2B1B0AF B6B5B4B3 BAB9B8B7 BEBDBCBB C2C1C0BF C6C5C4C3
CAC9C8C7 CECDCCCB D2D1D0CF D6D5D4D3 DAD9D8D7 DEDDDCDB E2E1E0DF E6E5E4E3
EAE9E8E7 EEEDECEB F2F1F0EF F6F5F4F3 FAF9F8F7 0100FCFB 05040302 09080706
0D0C0B0A 11100F0E 15141312 19181716 1D1C1B1A 21201F1E 25242322 29282726
2D2C2B2A 31302F2E 35343332 39383736 3D3C3B3A 41403F3E 45444342 49484746
4D4C4B4A 51504F4E 55545352 59585756 5D5C5B5A 61605F5E 65646362 69686766
6D6C6B6A 71706F6E 75747372 79787776 7D7C7B7A 81807F7E 85848382 89888786
8D8C8B8A 91908F8E 95949392 99989796 9D9C9B9A A1A09F9E A5A4A3A2 A9A8A7A6
ADACABAA B1B0AFAE B5B4B3B2 B9B8B7B6 BDBCBBBA C1C0BFBE C5C4C3C2 C9C8C7C6
CDCCBCBA D1D0CFCE D5D4D3D2 D9D8D7D6 DDDCDBDA E1E0DFDE E5E4E3E2 E9E8E7E6
EDECEBEA F1F0EFEE F5F4F3F2 F9F8F7F6 00FCFBFA 04030201 080706
```

client -> server:

```
write_sequence
00000000 00000001
header type
17 (application data)
major version
03
minor version
01
Length
03FB
data encrypted on client write key using parameters 1.2.643.7.1.2.5.1.1
(id-tc26-gost-28147-param-Z) in CNT mode
B105F95F 98A7E3B5 B9C9FA1D 6CECC918 118135C8 3F544257 2D6249EA C7B4C2DE
16E89EEE 9627378B 77CE9CDF 9AE08748 834C28C5 83D8F831 630036AD 0F45716D
CF40AE65 F370EF6A 8352197C 1C831D15 6DCF82CB F63EC447 A1FD7B9B AE16FC20
ACEB573E 8D74C088 02D39923 00A55B4D 0B0966F9 255AF13B 45C8D429 3AE34268
7C88A87B 8076EBB6 B68E4BA3 664538AA 76E4BFBE 3DA20CB2 E2749B3B 71E1CB9E
02AB330A 84FEABC7 7563E4BE 34F918EC 238D9479 7661B96A C2FBE919 4C68971C
F03E11D2 5F32DB37 FF8D7636 1C9C4C13 01AB3B33 595619BB 2772B3B7 6239F653
E8FB98BC CA533DC9 3C9A6AC3 3FF8C5BC E359573B C0349389 D08145C6 F7A6DBF4
DFDDC42C A822148C 437DDA5E 28275A31 A9277739 08BFBDDC 9294130F 40E36D72
22A2ACA1 17F57FB5 CBC4553D A0BE1EF0 F59D1753 9967C9F0 B1A2DEC3 D29F0E55
2612CA9C 8CF08BA9 C15D3C44 7013DC7B E70085CC 5F7BFB00 BB743013 4330BB51
AF31C999 90AADEA9 AEF1297 B89E2870 04233A87 44994E7D 493D84C6 E3CF3A50
5EC220C3 74050820 53E107F2 B34C38CA FF012FF5 DFD541E6 1F145230 C4AD73E1
7439F6C5 1E718C59 9A1504C5 F2209CEA CDBA7532 DE58A62F F8C215BE 04AEB7FB
0AA9B456 6616B9BA 8E082D4B BB0C44D7 5CCC28A6 1CCAE9CC 34D491E5 4B652B55
7CA7CAFB 9AE40199 AD963AE0 19595CE4 30FA2B78 B9F0A58F 974ED710 34346C28
4C6C5561 2111B6E2 8B481E6D 8504926C 6671810B D82FFF80 18BF728D A63A4AE7
8A9F16F6 500F6128 0B4501C6 7DACBD33 81FD7C98 6EA2683B B63F9CD9 2F59AD51
B732F1EC D34E594D C253DF3C 29F0A9B3 80EB50AA 35756475 28C1100E 2D6B6443
3826DB38 3EDE2819 A6D3E1E5 67401D7C C5B2E5CC 1ACEBB20 7BC76C7B 81E58A2C
FDE5C2D4 8D7D2021 AEA28287 9E01180C 32391F84 38BEF35A 16625049 34A6973A
F36A3E2C 971A37DE EF695B70 FF896284 7D358CFB A4978042 F38A88E6 786274F0
```



```

16 (handshake)
major version
03
minor version
01
Length
032F

message type
02 (server hello)
length
000046
TLS VERSION
    major version
    03
    minor version
    01

server random
52E78F3A 621D00F7 DFE637A4 179899F8 1570C14B B47D450B 53A474DC 7E4FFF19
session id length
20
session id
85FD2947 6E8E968D E34347F6 7736EFF1 12E3E149 3F7F1CA8 283BBE03 363A9F3A
ciphersuite
FF85 TLS_CIPHER_2012
compression method
00
extensions

message type
0B (certificate)
length
0002DD
server certificate chain
0002DA00 02D73082 02D33082 0282A003 02010202 0A427E80 21000000 8B927530
0806062A 85030202 03303A31 12301006 0A099226 8993F22C 64011916 02727531
12301006 0A099226 8993F22C 64011916 02637031 10300E06 03550403 13077465
73742D63 61301E17 0D313430 31323830 34323432 385A170D 32343031 32383131
30343238 5A304731 18301606 03550403 0C0F746C 73636F6E 665F7372 76353132
65312B30 2906092A 864886F7 0D010901 161C746C 73636F6E 665F7372 76353132
65406372 7970746F 70726F2E 72753066 301F0608 2A850307 01010101 30130607
2A850302 02240006 082A8503 07010102 02034300 0440442B 2744A488 9B1F456F
A23BC840 9FC272E8 F36A65EB 2548038B B0E10A8A FE2CF7B7 5AAAD185 D2F18B5A
ED28F747 B3C72D5C C5BA870A 2CC76F57 5A9BC7EF 25E7A382 01563082 01523013
0603551D 25040C30 0A06082B 06010505 07030130 0B060355 1D0F0404 03020430
301D0603 551D0E04 1604146B BF36F2E8 0DEB1254 4478CE0B 55CC7C84 E1B83430
1F060355 1D230418 30168014 9E03F0B8 9CFC60DC 8A181EE8 00DFA85B 32CD7376
303F0603 551D1F04 38303630 34A032A0 30862E68 7474703A 2F2F766D 2D746573
742D6361 2E63702E 72752F43 65727445 6E726F6C 6C2F7465 73742D63 612E6372
6C3081AC 06082B06 01050507 01010481 9F30819C 304B0608 2B060105 05073002
863F6874 74703A2F 2F766D2D 74657374 2D63612E 63702E72 752F4365 7274456E
726F6C6C 2F766D2D 74657374 2D63612E 63702E72 755F7465 73742D63 612E6372
74304D06 082B0601 05050730 02864166 696C653A 2F2F5C5C 766D2D74 6573742D
63612E63 702E7275 5C436572 74456E72 6F6C6C5C 766D2D74 6573742D 63612E63
702E7275 5F746573 742D6361 2E637274 30080606 2A850302 02030341 000FDCCB
2BEDB244 BB93D103 A9F058BA 46474261 300D87B8 6AD845AA B22CD64A B057AD90

```



```

client private key
E0C6D1EB 9AE63CF0 E07CA8C0 B6A04080 FBE89CCB B1C84997 8620D973 26696FC8
server public key blob header
06200100 492E0000 4D414731 00020000 30130607 2A850302 02240006 082A8503
07010102 02442B27
server public key
442B2744 A4889B1F 456FA23B C8409FC2 72E8F36A 65EB2548 038BB0E1 0A8AFE2C
FCB75AAA D185D2F1 8B5AED28 F747B3C7 2D5CC5BA 870A2CC7 6F575A9B C7EF25E7
hash(client random, server random)
50D55A4B B4D33355 987586D0 0F9CA5BA EB6CCD90 F64F5459 EBD90D23 DC40BDDE
ukm
50D55A4B B4D33355
Inner Point
50E03566 E44AC879 46640AF2 F5A286A7 2F9DF455 42BB60CB 152CE8EA 97BAF55D
4334089D F4E0FAF9 44D63A59 FABD132C 111D7924 AE96E25B 324D2F26 826D73AA
VKO result
2308FEBF 49F951F9 55650450 D90A4182 E12A48EF A932E8A0 214FBF8B 198DCE79
diversified key
7577FC0A 004E1801 9961BD3C 08C1E160 8BA355E3 02EDE8CA C627C6AA 86BC0026
premastersecret encrypted on exchange key using parameters
1.2.643.7.1.2.5.1.1 (id-tc26-gost-28147-param-Z) in ECB mode
      83BE410A 078B3130 50F47E89 124DE9DC EC591B77 0D0AB638
712E6F84 12A874BA
      MAC on exchange key using parameters
1.2.643.7.1.2.5.1.1 (id-tc26-gost-28147-param-Z) with ukm as IV
      6EE7685F

client key exchange
30433041 30280420 83BE410A 078B3130 50F47E89 124DE9DC EC591B77 0D0AB638
712E6F84 12A874BA 04046EE7 685FA015 06092A85 03070102 05010104 0850D55A
4BB4D333 55

client -> server:

header type
16 (handshake)
major version
03
minor version
01
length
032A

message type
0B (certificate)
length
0002DD
client certificate chain
0002DA00 02D73082 02D33082 0282A003 02010202 0A427E82 35000000 8B927630
0806062A 85030202 03303A31 12301006 0A099226 8993F22C 64011916 02727531
12301006 0A099226 8993F22C 64011916 02637031 10300E06 03550403 13077465
73742D63 61301E17 0D313430 31323830 34323432 395A170D 32343031 32383131
30343239 5A304731 18301606 03550403 0C0F746C 73636F6E 665F636C 6E353132
65312B30 2906092A 864886F7 0D010901 161C746C 73636F6E 665F636C 6E353132
65406372 7970746F 70726F2E 72753066 301F0608 2A850307 01010101 30130607
2A850302 02240006 082A8503 07010102 02034300 0440EBCE 8AB9C7C8 400ED94A

```



```

client public key blob header
06200100 492E0000 4D414731 00020000 30130607 2A850302 02240006 082A8503
07010102 02EBCE8A
client public key
EBCE8AB9 C7C8400E D94AC055 529EDA2B 31102615 BD632252 27E52FF8 D9E86E6F
7D1FBA5E 292DEECE D28BAB6F 8B82D34F B71B2457 3838D279 E9DBF1D9 AC711AD9
hash(client random, server random)
50D55A4B B4D33355 987586D0 0F9CA5BA EB6CCD90 F64F5459 EBD90D23 DC40BDDE
ukm
50D55A4B B4D33355
premastersecret encrypted on exchange key using parameters
1.2.643.7.1.2.5.1.1(id-tc26-gost-28147-param-Z) in ECB mode
83BE410A 078B3130 50F47E89 124DE9DC EC591B77 0D0AB638
712E6F84 12A874BA
MAC on exchange key using parameters
1.2.643.7.1.2.5.1.1 (id-tc26-gost-28147-param-Z) with ukm as IV
6EE7685F
Inner Point
50E03566 E44AC879 46640AF2 F5A286A7 2F9DF455 42BB60CB 152CE8EA 97BAF55D
4334089D F4E0FAF9 44D63A59 FABD132C 111D7924 AE96E25B 324D2F26 826D73AA
VKO result
2308FEBF 49F951F9 55650450 D90A4182 E12A48EF A932E8A0 214FBF8B 198DCE79
diversified key
7577FC0A 004E1801 9961BD3C 08C1E160 8BA355E3 02EDE8CA C627C6AA 86BC0026
premastersecret
9F82C8A2 D3A235C9 DEBE5E34 C1D5AE45 88B77B21 C3051BFD 3FAB5546 9D09C124

master secret = PRF_GOSTR3411(premastersecret, "master secret", client
random, server random)
57380114 BB1FA165 D40DF7CF C56375FE F23D80FF 06000A53 FBA6D596 50C31B11
98B2808B 2E2A623C 4B546B26 0428DBA2
hash(master secret)
1386F6F2 946BE70C 02AA1777 BF8A862F 8CB15046 FC9E35EA F16D4D09 5464D5FE
key_block = PRF_GOSTR3411(hash(master secret), "key expansion", server
random, client random)
A5E1F809 F154CF9C DE7C3300 398D1EAF 1CDE3F35 B1572301 AA423DFA CBD60936
1563C9CD FEBC7DFE 42AD43DA 3322F0D3 5F52D230 27737CF6 8AE46510 F2FDC59B
4A2A9D2C 4FAF9BFE 05D2A004 FC245911 2D66AC99 9CF6BE7D 333FE44F 213C3F86
5ACDE82E 844CFE5C 2141C51D 16C96784 DCBEAB64 FC952146 575CFA3A EB4EC8CB
47AA2337 EB8BFC56 400CC991 58A3E03E

server read key
4A2A9D2C 4FAF9BFE 05D2A004 FC245911 2D66AC99 9CF6BE7D 333FE44F 213C3F86
server read MAC secret
A5E1F809 F154CF9C DE7C3300 398D1EAF 1CDE3F35 B1572301 AA423DFA CBD60936
server read IV
47AA2337 EB8BFC56

client <- server:

header
header type
14 (change cipher spec)
major version
03
minor version

```


client internals:

client read key

5ACDE82E 844CFE5C 2141C51D 16C96784 DCBEAB64 FC952146 575CFA3A EB4EC8CB

client read MAC secret

1563C9CD FEBC7DFE 42AD43DA 3322F0D3 5F52D230 27737CF6 8AE46510 F2FDC59B

client read IV

400CC991 58A3E03E

Client message

04030201 08070605 0C0B0A09 100F0E0D 14131211 18171615 1C1B1A19 201F1E1D
24232221 28272625 2C2B2A29 302F2E2D 34333231 38373635 3C3B3A39 403F3E3D
44434241 48474645 4C4B4A49 504F4E4D 54535251 58575655 5C5B5A59 605F5E5D
64636261 68676665 6C6B6A69 706F6E6D 74737271 78777675 7C7B7A79 807F7E7D
84838281 88878685 8C8B8A89 908F8E8D 94939291 98979695 9C9B9A99 A09F9E9D
A4A3A2A1 A8A7A6A5 ACABAAA9 B0AFAEAD B4B3B2B1 B8B7B6B5 BCBBBAB9 C0BFBEBD
C4C3C2C1 C8C7C6C5 CCCBCAC9 D0CFCECD D4D3D2D1 D8D7D6D5 DCDBDAD9 E0DFDEDD
E4E3E2E1 E8E7E6E5 ECEBEAE9 F0EFEFEE F4F3F2F1 F8F7F6F5 FCFBFAF9 03020100
07060504 0B0A0908 0F0E0D0C 13121110 17161514 1B1A1918 1F1E1D1C 23222120
27262524 2B2A2928 2F2E2D2C 33323130 37363534 3B3A3938 3F3E3D3C 43424140
47464544 4B4A4948 4F4E4D4C 53525150 57565554 5B5A5958 5F5E5D5C 63626160
67666564 6B6A6968 6F6E6D6C 73727170 77767574 7B7A7978 7F7E7D7C 83828180
87868584 8B8A8988 8F8E8D8C 93929190 97969594 9B9A9998 9F9E9D9C A3A2A1A0
A7A6A5A4 ABAAA9A8 AFAEADAC B3B2B1B0 B7B6B5B4 BBBAB9B8 BFBEBDBC C3C2C1C0
C7C6C5C4 CBCAC9C8 CFCECDCC D3D2D1D0 D7D6D5D4 DBDAD9D8 DFDEDDDC E3E2E1E0
E7E6E5E4 EBEBE9E8 EFEEEDDEC F3F2F1F0 F7F6F5F4 FBFAF9F8 020100FC 06050403
0A090807 0E0D0C0B 1211100F 16151413 1A191817 1E1D1C1B 2221201F 26252423
2A292827 2E2D2C2B 3231302F 36353433 3A393837 3E3D3C3B 4241403F 46454443
4A494847 4E4D4C4B 5251504F 56555453 5A595857 5E5D5C5B 6261605F 66656463
6A696867 6E6D6C6B 7271706F 76757473 7A797877 7E7D7C7B 8281807F 86858483
8A898887 8E8D8C8B 9291908F 96959493 9A999897 9E9D9C9B A2A1A09F A6A5A4A3
AAA9A8A7 AEADACAB B2B1B0AF B6B5B4B3 BAB9B8B7 BEBDBCBB C2C1C0BF C6C5C4C3
CAC9C8C7 CEDCCCB D2D1D0CF D6D5D4D3 DAD9D8D7 DEDDDCDB E2E1E0DF E6E5E4E3
EAE9E8E7 EEEDECEB F2F1F0EF F6F5F4F3 FAF9F8F7 0100FCFB 05040302 09080706
0D0C0B0A 11100F0E 15141312 19181716 1D1C1B1A 21201F1E 25242322 29282726
2D2C2B2A 31302F2E 35343332 39383736 3D3C3B3A 41403F3E 45444342 49484746
4D4C4B4A 51504F4E 55545352 59585756 5D5C5B5A 61605F5E 65646362 69686766
6D6C6B6A 71706F6E 75747372 79787776 7D7C7B7A 81807F7E 85848382 89888786
8D8C8B8A 91908F8E 95949392 99989796 9D9C9B9A A1A09F9E A5A4A3A2 A9A8A7A6
ADACABAA B1B0FAE B5B4B3B2 B9B8B7B6 BDBCBBBA C1C0BFBE C5C4C3C2 C9C8C7C6
CDCCBCA D1D0CFCE D5D4D3D2 D9D8D7D6 DDDCDBDA E1E0DFDE E5E4E3E2 E9E8E7E6
EDECEBEA F1F0EFEE F5F4F3F2 F9F8F7F6 00FCFBFA 04030201 080706

client -> server:

write_sequence

00000000 00000001

header type

17 (application data)

major version

03

minor version

01

length

03FB


```
0002
data encrypted on client write key using parameters 1.2.643.7.1.2.5.1.1
(id-tc26-gost-28147-param-Z) in CNT mode
F835
MAC on client write MAC secret using parameters 1.2.643.7.1.2.5.1.1 (id-
tc26-gost-28147-param-Z)
E1236D00
```

A.3. TLS 1.0 аутентификация клиента с ключом 256 бит и сервера с ключом 512 бит

```
client -> server:
```

```
header type
16 (handshake)
major version
03
minor version
01
Length
004F

message type
01 (client hello)
length
00004B
TLS VERSION
                                major version
                                03
                                minor version
                                01

client random
52E78EFE 6E681041 EC766E3D E0B54F24 3AE4C48C 5CE47EEE 84FBDA38 F5C50D64
session id length
00
session id

ciphersuite length
0004
ciphersuite
0081          TLS_CIPHER_2001
FF85          TLS_CIPHER_2012
compression methods
                                length
                                01
                                compression method
                                00

extensions
  extensions length
  001E
  renegotiation info (FF01)
  length
  0001
  Renegotiation Info extension
  00
```


65406372 7970746F 70726F2E 72753066 301F0608 2A850307 01010101 30130607
2A850302 02240006 082A8503 07010102 02034300 0440EBCE 8AB9C7C8 400ED94A
C055529E DA2B3110 2615BD63 225227E5 2FF8D9E8 6E6F7D1F BA5E292D EECED28B
AB6F8B82 D34FB71B 24573838 D279E9DB F1D9AC71 1AD9A382 01563082 01523013
0603551D 25040C30 0A06082B 06010505 07030230 0B060355 1D0F0404 03020430
301D0603 551D0E04 16041483 51A75EF8 F9450E71 DC49382D 2D3BD3F0 0E5ABF30
1F060355 1D230418 30168014 9E03F0B8 9CFC60DC 8A181EE8 00DFA85B 32CD7376
303F0603 551D1F04 38303630 34A032A0 30862E68 7474703A 2F2F766D 2D746573
742D6361 2E63702E 72752F43 65727445 6E726F6C 6C2F7465 73742D63 612E6372
6C3081AC 06082B06 01050507 01010481 9F30819C 304B0608 2B060105 05073002
863F6874 74703A2F 2F766D2D 74657374 2D63612E 63702E72 752F4365 7274456E
726F6C6C 2F766D2D 74657374 2D63612E 63702E72 755F7465 73742D63 612E6372
74304D06 082B0601 05050730 02864166 696C653A 2F2F5C5C 766D2D74 6573742D
63612E63 702E7275 5C436572 74456E72 6F6C6C5C 766D2D74 6573742D 63612E63
702E7275 5F746573 742D6361 2E637274 30080606 2A850302 02030341 00850B51
588B2C0E 4C9AB975 778130DF E3F6D654 D7326F90 370A3C38 A2470D5C 8CB91E5B
580BA724 418CC995 7A3BF2F3 1DA5B8DB 3A0091A0 EE150FCB CFBE2EB 2B

premastersecret

26DBE1DA A8757A2F FD12E2BB 1ABA62CC A69C37B1 80C12B7D 8FEF63AC 17723A25

client private key

E0C6D1EB 9AE63CF0 E07CA8C0 B6A04080 FBE89CCB B1C84997 8620D973 26696FC8

client private key (ephemeral)

4CE22FCB 57E076BB B68E3231 C30D6B93 FA388247 78C7179C 4A668BF4 B9A0BA57

1D0EA2DA D4DD03A8 BAF18AEA 3FAE5E41 DFDA273C 4E63948D DC1A06F9 D61A869E

server public key blob header

06200100 3D2E0000 4D414731 00040000 30150609 2A850307 01020102 0106082A

85030701 010203F0

server public key

F0DBBD44 A22D19AF CB22A0CA D421A02E 7930D5C6 E549C13B BF7D1437 7C67CAE8

7D45E79E 27E96D63 1A0AA6C7 1E6B353C 66A02362 F9D43FC3 F53DDFAB 567CDFB9

2909DCB6 2A931896 ED1F1D16 55AC584D C8D6745C 51EA68CA C8EBF49C AC305EFA

64285097 99F30D21 9A840782 7EB27629 3E7BC1AE 68DCB939 FAD9BFBC AD938C84

hash(client random, server random)

F528F112 8286A80F 19405393 1267A151 17455680 C30C1600 84B400B3 867208B2

ukm

F528F112 8286A80F

Inner Point

02756997 5C3DEF17 6C9407B4 D26A87D0 35BAA977 74EE19D0 B7B37109 2BE9B3C7

9C9174A1 9371CD74 F4C86ECF 9BD5952D AE967D91 B44FB899 D0608B68 753F6604

81317270 F68976BD 893A9D6E FE942CF3 6A002385 5F76B986 C4FD8449 CFBC56D9

9F91E5C1 DD62778D FAAB3640 DDB4E7E0 4DEC3FDF CECBA04D 978CAB7C 7689E5EF

VKO result

8A2D3569 221F2B2B 3207450E 9D32E9D0 973ED97E 67E96D5B A06131DF 4DA75AA8

diversified key

D83FE5FA 27803F24 AB15153D 4CCDB351 343B6BE3 CCAAAA29 E2CCC38E CB771524

premastersecret encrypted on exchange key using parameters

1.2.643.7.1.2.5.1.1 (id-tc26-gost-28147-param-Z) in ECB mode

2B9733F1 F6EFEB45 30354151 19E46D3E 1798A037 488BE6B5

836CF8CF B81BB597

MAC on exchange key using parameters

1.2.643.7.1.2.5.1.1 (id-tc26-gost-28147-param-Z) with ukm as IV

E2897619

client key exchange

3081F230 81EF3028 04202B97 33F1F6EF EB453035 415119E4 6D3E1798 A037488B

```
E6B5836C F8CFB81B B5970404 E2897619 A081C206 092A8503 07010205 0101A081
AA302106 082A8503 07010101 02301506 092A8503 07010201 02010608 2A850307
01010203 03818400 04818094 505C443B 6C2685C5 FD88888F 00177F56 2F28FEB2
6461389D 2BC70730 B94B903D 526103DA AF2F360B 47EEDCC4 BA07F733 D83D33AF
0CA81936 8C3C5F08 26A67D44 14BF92A2 4A89C784 1B81C101 674EDC77 1D092BEA
46ACD877 C60F7DA1 60B959A1 428A2F72 D2DDDE4B 741D79E5 9878F02E 12241798
1C67A557 4CEF3F48 9E679504 08F528F1 128286A8 0F
```

hash(handshake messages)

```
76DC62A9 89673B8D E1B4C947 600DDE62 6383CBB7 3A6BCC53 9D66F7B0 42AB654A
```

certificate verify

```
F17A142A 4D306C61 1C5FE211 055A40A1 18A3080F 2A82771F 9447E4C9 A5AEC9C6
4411AC5A BDBDBA1F 6FBD23CD 3AC0873C ED7B0C90 8CAFDB07 60E8B20E DEA72A9F
```

client -> server:

header type

16 (handshake)

major version

03

minor version

01

Length

041E

message type

0B (certificate)

length

02DD

client certificate chain

```
0002DA00 02D73082 02D33082 0282A003 02010202 0A427E82 35000000 8B927630
0806062A 85030202 03303A31 12301006 0A099226 8993F22C 64011916 02727531
12301006 0A099226 8993F22C 64011916 02637031 10300E06 03550403 13077465
73742D63 61301E17 0D313430 31323830 34323432 395A170D 32343031 32383131
30343239 5A304731 18301606 03550403 0C0F746C 73636F6E 665F636C 6E353132
65312B30 2906092A 864886F7 0D010901 161C746C 73636F6E 665F636C 6E353132
65406372 7970746F 70726F2E 72753066 301F0608 2A850307 01010101 30130607
2A850302 02240006 082A8503 07010102 02034300 0440EBCE 8AB9C7C8 400ED94A
C055529E DA2B3110 2615BD63 225227E5 2FF8D9E8 6E6F7D1F BA5E292D EECED28B
AB6F8B82 D34FB71B 24573838 D279E9DB F1D9AC71 1AD9A382 01563082 01523013
0603551D 25040C30 0A06082B 06010505 07030230 0B060355 1D0F0404 03020430
301D0603 551D0E04 16041483 51A75EF8 F9450E71 DC49382D 2D3BD3F0 0E5ABF30
1F060355 1D230418 30168014 9E03F0B8 9CFC60DC 8A181EE8 00DFA85B 32CD7376
303F0603 551D1F04 38303630 34A032A0 30862E68 7474703A 2F2F766D 2D746573
742D6361 2E63702E 72752F43 65727445 6E726F6C 6C2F7465 73742D63 612E6372
6C3081AC 06082B06 01050507 01010481 9F30819C 304B0608 2B060105 05073002
863F6874 74703A2F 2F766D2D 74657374 2D63612E 63702E72 752F4365 7274456E
726F6C6C 2F766D2D 74657374 2D63612E 63702E72 755F7465 73742D63 612E6372
74304D06 082B0601 05050730 02864166 696C653A 2F2F5C5C 766D2D74 6573742D
63612E63 702E7275 5C436572 74456E72 6F6C6C5C 766D2D74 6573742D 63612E63
702E7275 5F746573 742D6361 2E637274 30080606 2A850302 02030341 00850B51
588B2C0E 4C9AB975 778130DF E3F6D654 D7326F90 370A3C38 A2470D5C 8CB91E5B
580BA724 418CC995 7A3BF2F3 1DA5B8DB 3A0091A0 EE150FCB CFBEB2EB 2B
```

message type

10 (client key exchange)

length


```

client public key (ephemeral)
94505C44 3B6C2685 C5FD8888 8F00177F 562F28FE B2646138 9D2BC707 30B94B90
3D526103 DAAF2F36 0B47EEDC C4BA07F7 33D83D33 AF0CA819 368C3C5F 0826A67D
4414BF92 A24A89C7 841B81C1 01674EDC 771D092B EA46ACD8 77C60F7D A160B959
A1428A2F 72D2DDDE 4B741D79 E59878F0 2E122417 981C67A5 574CEF3F 489E6795
hash(client random, server random)
F528F112 8286A80F 19405393 1267A151 17455680 C30C1600 84B400B3 867208B2
ukm
F528F112 8286A80F
premastersecret encrypted on exchange key using parameters
1.2.643.7.1.2.5.1.1(id-tc26-gost-28147-param-Z) in ECB mode
      2B9733F1 F6EFEB45 30354151 19E46D3E 1798A037 488BE6B5
836CF8CF B81BB597
      MAC on exchange key using parameters
1.2.643.7.1.2.5.1.1 (id-tc26-gost-28147-param-Z) with ukm as IV
      E2897619

Inner Point
02756997 5C3DEF17 6C9407B4 D26A87D0 35BAA977 74EE19D0 B7B37109 2BE9B3C7
9C9174A1 9371CD74 F4C86ECF 9BD5952D AE967D91 B44FB899 D0608B68 753F6604
81317270 F68976BD 893A9D6E FE942CF3 6A002385 5F76B986 C4FD8449 CFBC56D9
9F91E5C1 DD62778D FAAB3640 DDB4E7E0 4DEC3FDF CECBA04D 978CAB7C 7689E5EF
VKO result
8A2D3569 221F2B2B 3207450E 9D32E9D0 973ED97E 67E96D5B A06131DF 4DA75AA8
diversified key
D83FE5FA 27803F24 AB15153D 4CCDB351 343B6BE3 CCAAAA29 E2CCC38E CB771524
premastersecret
26DBE1DA A8757A2F FD12E2BB 1ABA62CC A69C37B1 80C12B7D 8FEF63AC 17723A25
client public key blob header
06200100 492E0000 4D414731 00020000 30130607 2A850302 02240006 082A8503
07010102 02EBCE8A
client public key
EBCE8AB9 C7C8400E D94AC055 529EDA2B 31102615 BD632252 27E52FF8 D9E86E6F
7D1FBA5E 292DEECE D28BAB6F 8B82D34F B71B2457 3838D279 E9DBF1D9 AC711AD9

master secret = PRF_GOSTR3411(premastersecret, "master secret", client
random, server random)
A9953B3D 2F36CEF7 600E6068 171E1C75 99164F6E 3FCC4321 F77A9AB4 963D2AFA
41A54CF9 8FCFAE40 ADF42F8F E668FD75
hash(master secret)
56E69592 B7D44D4D 0D5A00A7 F8CDA91D C2AE886E 34C856B4 E5B98BB9 1CD5459E
key_block = PRF_GOSTR3411(hash(master secret), "key expansion", server
random, client random)
B826AA51 EB0A4DEB 0BF5689D 2C7E7997 6744B69C 9F16DEF4 77BFF655 023F4964
831B5706 AC8AB7C5 1BD12D17 40D61837 97F6874E 31EFED44 A1C64BDB D6D3AED2
4F0520AA D7A6825F D2D73C61 5801EB0E CC41644F 8FF488EB 34635D33 7644BF76
328D89F9 36B29ACC 703CE9B5 203149BD BD0BED66 D896B5EE 08D4CB8D 90D99E66
34F71DD0 409DFB54 956B327C C1B606FE

server read key
4F0520AA D7A6825F D2D73C61 5801EB0E CC41644F 8FF488EB 34635D33 7644BF76
server read MAC secret
B826AA51 EB0A4DEB 0BF5689D 2C7E7997 6744B69C 9F16DEF4 77BFF655 023F4964
server read IV
34F71DD0 409DFB54

client <- server:

```



```
header type
17 (application data)
major version
03
minor version
01
Length
02
data encrypted on client write key using parameters 1.2.643.7.1.2.5.1.1
(id-tc26-gost-28147-param-Z) in CNT mode
6016
MAC on client write MAC secret using parameters 1.2.643.7.1.2.5.1.1 (id-
tc26-gost-28147-param-Z)
C477AD86
```

A.4. TLS 1.0 аутентификация сервера с ключом 256 бит

```
client -> server:
```

```
header type
16 (handshake)
major version
03
minor version
01
Length
004E

message type
01 (client hello)
length
00004A
TLS VERSION
                                major version
                                03
                                minor version
                                01

client random
52E78F4F 6AA0FE31 2217AEF6 91AD7639 32945E8C EDD7F96E 3C336B08 66A66698
session id length
00
session id

ciphersuite length
0004
CipherSuite
0081          TLS_CIPHER_2001
FF85          TLS_CIPHER_2012
compression methods
                                length
                                01
                                compression method
                                00

extensions
  extensions length
```



```
write_sequence
00000000 00000002
header type
17 (application data)
major version
03
minor version
01
length
0002
data encrypted on client write key using parameters 1.2.643.7.1.2.5.1.1
(id-tc26-gost-28147-param-Z) in CNT mode
4A03
MAC on client write MAC secret using parameters 1.2.643.7.1.2.5.1.1 (id-
tc26-gost-28147-param-Z)
D913A774
```

Приложение В Модули ASN.1

```
Gost-CryptoPro-TLS
Gost-CryptoPro-TLS { iso(1) member-body(2) ru(643) rans(2)
    cryptopro(2) other(1) modules(1) gost-CryptoPro-TLS(16) 1 }

DEFINITIONS ::=
BEGIN
-- EXPORTS ALL --
-- Типы и значения, определенные в этом модуле, экспортируются для
-- использования в других модулях ASN.1, содержащихся в российских
-- криптографических спецификациях «ГОСТ» и «ГОСТ Р», и для использования
-- другими приложениями для получения доступа к российским службам
-- криптографии. Другие приложения могут использовать их в своих целях, но
-- это не накладывает ограничений на расширения и изменения, необходимые
-- для поддержания или улучшения российских служб криптографии.

IMPORTS

Certificate, AlgorithmIdentifier
FROM PKIX1Explicit88 {iso(1) identified-organization(3) dod(6)
    internet(1) security(5) mechanisms(5) pkix(7) id-mod(0)
    id-pkix1-explicit-88(1)}

id-CryptoPro-algorithms, gostR3410-EncryptionSyntax
FROM Cryptographic-Gost-Useful-Definitions { iso(1) member-body(2)
    ru(643) rans(2) cryptopro(2) other(1) modules(1)
    cryptographic-Gost-Useful-Definitions(0) 1 }

GostR3410-KeyTransport
FROM GostR3410-EncryptionSyntax gostR3410-EncryptionSyntax;

id-PRF-GostR3411-94 OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms prf-gostr3411-94(23) }

TLSProxyKeyTransportBlob ::= SEQUENCE {
    keyBlob          GostR3410-KeyTransport,
    cert             OCTET STRING
}

TLSGostKeyTransportBlob ::= SEQUENCE {
    keyBlob          GostR3410-KeyTransport,
    proxyKeyBlobs   SEQUENCE OF TLSProxyKeyTransportBlob OPTIONAL
}

TLSGostSrvKeyExchange ::= SEQUENCE OF OCTET STRING (CONSTRAINED BY {
    Certificate })

TLSGostExtensionHashHMACSelect ::= SEQUENCE {
    hashAlgorithm    AlgorithmIdentifier,
    hmacAlgorithm    AlgorithmIdentifier,
    prfAlgorithm     AlgorithmIdentifier
}
```



```
TLSGostExtensionHashHMACSelectClient ::=
    SEQUENCE OF TLSGostExtensionHashHMACSelect

TLSGostExtensionHashHMACSelectServer ::= TLSGostExtensionHashHMACSelect

END -- Gost-CryptoPro-TLS
```

Библиография

[**IETF RFC 4357**] В. Попов, И. Курепкин и 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), RFC 4357, январь 2006 г.

[**IETF RFC 4490**] Под ред. С. Леонтьева и Г. Чудова «Использование алгоритмов ГОСТ 28147-89, ГОСТ Р 34.11-94, ГОСТ Р 34.10-94 и ГОСТ Р 34.10-2001 с синтаксисом криптографических сообщений (CMS)» (Leontiev, S., Ed. and G. Chudov, Ed., Using the GOST 28147-89, GOST R 34.11-94, GOST R 34.10-94, and GOST R 34.10-2001 Algorithms with Cryptographic Message Syntax (CMS)), RFC 4490, май 2006.

[**IETF RFC 4491**] Под ред. С. Леонтьева и Д. Шефановского «Использование алгоритмов по ГОСТ Р 34.10-94, ГОСТ Р 34.10-2001 и ГОСТ Р 34.11-94 в профиле сертификата и списка отзыва сертификатов (CLR) инфраструктуры открытых ключей Интернет X.509» (Leontiev, S., Ed. and D. Shefanovskij, Ed., Using the GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms with the Internet X.509 Public Key Infrastructure Certificate and CRL Profile), RFC 4491, май 2006.

[**IETF RFC 5216**] Д. Симон, Б. Абоба и Р. Харст, «Протокол аутентификации EAP-TLS» (Simon D., Aboba B. and R. Hurst, The EAP-TLS Authentication Protocol), RFC 5216, март 2008 г.