Abstract

This document is intended to register new cipher suites for the Transport Layer Security (TLS) protocol, according to the procedure specified in TLS Protocol standards. These cipher suites are based on Russian national cryptographic standards - GOST R 34.10-94 and GOST R 34.10-2001 public keys, GOST 28147-89 encryption algorithm and GOST R 34.11-94 digest algorithm.
Table of Contents

1. Introduction ................................................. 3
2. CipherSuite Definitions ...................................... 3
   2.1. Key Exchange ........................................... 3
   2.2. PRF, Signature and Hash ................................. 4
   2.3. Cipher and MAC ......................................... 4
3. Data Structures and Computations ............................... 5
   3.1. Algorithms ............................................. 5
   3.2. Keys Calculation ....................................... 5
   3.3. Server Certificate ..................................... 5
   3.4. Server Key Exchange ................................... 5
   3.5. Certificate Request .................................... 6
   3.6. Client Key Exchange Message ............................ 6
   3.7. Certificate Verify ...................................... 8
   3.8. Finished ................................................ 9
4. Compatibility .................................................. 9
5. Security Considerations ....................................... 9
6. IANA Considerations .......................................... 9
7. References ................................................... 10
   7.1. Normative references .................................... 10
   7.2. Informative references .................................. 11
Appendix A. ASN.1 Modules ........................................ 12
   A.1. Gost-CryptoPro-TLS ...................................... 12
Appendix B. Acknowledgments ...................................... 13
Authors’ Addresses .............................................. 15
Intellectual Property and Copyright Statements .................. 16
1. Introduction

This document proposes the addition of new cipher suites to the Transport Layer Security (TLS) protocol to support GOST R 34.11-94 digest, GOST 28147-89 encryption and VKO GOST R 34.10-94/2001 key exchange algorithms. The cipher suites defined here were proposed by CRYPTO-PRO Company for the "Russian Cryptographic Software Compatibility Agreement" community.

Algorithms GOST R 34.10-94, GOST R 34.10-2001, GOST 28147-89 and GOST R 34.11-94 have been developed by Russian Federal Agency of Governmental Communication and Information (FAGCI) and "All-Russian Scientific and Research Institute of Standardization". They are described in \[GOSTR341094\], \[GOSTR341001\], \[GOSTR341194\] and \[GOST28147\] \((GOST3431095), \[GOST3431004\], \[GOST3431195\]).

Algorithms VKO GOST R 34.10-94/2001 and PRF\_GOSTR3411 are described in \[CPALGS\].

This document defines two configurations:
- anonymous client - authenticated server (only server provides a certificate);
- authenticated client - authenticated server (client and server exchange certificates).

The presentation language used here is the same as in \[TLS1.2\]. Since this specification extends \[TLS1.2\], these descriptions should be merged with those in the TLS specification and any others that extend TLS. This means, that enum types may not specify all possible values and structures with multiple formats chosen with a select() clause may not indicate all possible cases.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in \[RFC2119\].

2. CipherSuite Definitions

2.1. Key Exchange

The cipher suites defined here use the following key exchange algorithms:
Key derivation algorithms based on GOST R 34.10-94 and GOST R 34.10-2001 public keys (VKO GOST R 34.10-94, VKO GOST R 34.10-2001) are described in [CPALGS].

### 2.2. PRF, Signature and Hash

For a PRF, described in section 5 of [TLS1.2], the cipher suites described here use PRF_GOSTR3411 (refer to Section 3.1). The same PRF MUST be used for all dependent protocols, such as [EAP-TLS].

GOST R 34.10-94/2001 signature is used for CertificateVerify message.

GOST R 34.11 digest algorithm ([GOSTR341194]) is used for CertificateVerify.signature.gostR3411_hash and Finished.verify_data (see sections 7.4.10 and 7.4.11 of [TLS1.2]).

### 2.3. Cipher and MAC

The following cipher algorithm and MAC functions are used (for details refer to Section 3.1):

<table>
<thead>
<tr>
<th>CipherSuite</th>
<th>Cipher</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS_GOSTR341094_WITH_28147_CNTIMIT</td>
<td>GOST28147</td>
<td>IMIT_GOST28147</td>
</tr>
<tr>
<td>TLS_GOSTR341001_WITH_28147_CNTIMIT</td>
<td>GOST28147</td>
<td>IMIT_GOST28147</td>
</tr>
<tr>
<td>TLS_GOSTR341094_WITH_NULL_GOSTR3411</td>
<td>-</td>
<td>HMAC_GOSTR3411</td>
</tr>
<tr>
<td>TLS_GOSTR341001_WITH_NULL_GOSTR3411</td>
<td>-</td>
<td>HMAC_GOSTR3411</td>
</tr>
</tbody>
</table>

For all four cipher suites, the use of MAC is slightly different from the one, described in section 6.2.3.1 of [TLS1.2]. In [TLS1.2], MAC is calculated from the following data:

\[
\text{MACed_data[seq_num]} = \text{seq_num} + \\
\text{TLSCompressed.type} + \\
\text{TLSCompressed.version} + \\
\text{TLSCompressed.length} + \\
\text{TLSCompressed.fragment};
\]
These cipher suites use the same input for first record, but for each next record the input from all previous records is concatenated:

$$\text{MACed\_data}[0] + \ldots + \text{MACed\_data}[n]$$

3. Data Structures and Computations

3.1. Algorithms

GOST 28147-89 [GOST28147] uses 256-bit key size and 8-byte IV. Cipher suites, defined here, use GOST 28147-89 as a stream cipher in counter mode with S-box from id-Gost28147-89-CryptoPro-A-ParamSet (see [CPALGS]) and CryptoPro key meshing algorithm.

IMIT_GOST28147 is GOST 28147-89 [GOST28147] in "IMITOVSTAVKA" mode (4 bytes)

HMAC_GOSTR3411(secret, data) is based on GOST R 34.11-94 digest and described in [CPALGS].

PRF_GOSTR3411(secret, label, seed) is based on HMAC_GOSTR3411 and described in [CPALGS].

3.2. Keys Calculation

Key calculation is done according to section 6.3 of [TLS1.2], with PRF_GOSTR3411 function used instead of PRF. The parameters are as follows:

- $\text{SecurityParameters.hash\_size} = 32$
- $\text{SecurityParameters.key\_material\_length} = 32$
- $\text{SecurityParameters.IV\_size} = 8$

Length of necessary key material is 144 bytes.

3.3. Server Certificate

For these cipher suites this message is required and it MUST contain a certificate, with a public key algorithm matching ServerHello.cipher\_suite.

3.4. Server Key Exchange

This message MUST NOT be used in these cipher suites, because all the parameters necessary are present in server certificate (see [CPPK]).
3.5. Certificate Request

This message is used as described in section 7.4.5 of [TLS1.2], and extended as follows:

```
enum {
    gostr341094(21), gostr34102001(22), (255)
} ClientCertificateType;
```

gostr341094 and gostr34102001 certificate types identify that the server accepts GOST R 34.10-94 and GOST R 34.10-2001 public key certificates.

[IANA please remove] Note: The above numeric definitions for ClientCertificateType have not yet been registered.

```
enum{
    gostr3411(XX), (255)
} HashType;
```

gostr3411 hash type identifies that the server accepts GOST R 34.11-94 hash function. It is RECOMMENDED to populate CertificateRequest.certificate_hash only with gostr3411 value, when one of the cipher suites described in this document is chosen.

3.6. Client Key Exchange Message

This message is used as described in section 7.4.9 of [TLS1.2], it is required for these suites, and contains DER-encoded TLSGostKeyTransportBlob structure [X.660].

```
enum { vko_gost } KeyExchangeAlgorithm;
```

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

ASN1-syntax for this structure is:
TLSGostKeyTransportBlob ::= SEQUENCE {
  keyBlob GostR3410-KeyTransport,
  proxyKeyBlobs SEQUENCE OF TLSProxyKeyTransportBlob OPTIONAL
}

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

GostR3410-KeyTransport is defined in [CPCMS].

keyBlob.transportParameters MUST be present.

keyBlob.transportParameters.ephemeralPublicKey MUST be present if the
server didn’t request client certificate or client’s public key
algorithm and parameters do not match those of the recipient. Else
it SHOULD be omitted.

proxyKeyBlobs - (optional) contains key exchange for secondary
recipients (for example, for the firewall, which audits connections).

cert - contains secondary recipient’s certificate.

Actions of client:

First, the client generates a random 32-byte premaster_secret.

Then shared_ukm is calculated as first 8 bytes of digest of
concatenated client random and server random:

shared_ukm = GOSTR3411(client_random|server_random)[0..7]

Then client chooses a sender key. If
keyBlob.transportParameters.ephemeralPublicKey is present, the
corresponding secret key MUST be used as a sender key. If it is
missing, the secret key, corresponding to the client certificate MUST
be used.

Using the sender key and recipient’s public key, algorithm VKO
GOST R 34.10-94 or VKO GOST R 34.10-2001 (described in [CPALGS]) is
applied to produce KEK. VKO GOST R 34.10-2001 is used with
shared_ukm as UKM.

Then CryptoPro Key Wrap algorithm is applied to encrypt
premaster_secret and produce CEK_ENC and CEK_MAC. Again, shared_ukm
is used as URM. keyBlob.transportParameters.encryptionParamSet is
used for all encryption operations.
The resulting encrypted key (CEK_ENC) is placed in keyBlob.sessionEncryptedKey.encryptedKey field, its mac (CEK_MAC) is placed in keyBlob.sessionEncryptedKey.macKey field, and shared_ukm (UKM) is placed in keyBlob.transportParameters.ukm field.

Actions of server:

Server MUST verify, that keyBlob.transportParameters.ukm is equal to GOSTR3411(client_random|server_random)[0..7], before decrypting the premaster_secret.

Server applies VKO GOST R 34.10-94 or VKO GOST R 34.10-2001, (depending on the client public key type), and CryptoPro Key Unwrap algorithm in the similar manner to decrypt the premaster_secret.

Server MUST verify keyBlob.sessionEncryptedKey.macKey after decrypting the premaster_secret.

3.7. Certificate Verify

This message is used as described in section 7.4.10 of [TLS1.2]. If the client have sent both a client certificate and an ephemeral public key, it MUST send a certificate verify message, as a proof of possession of the private key for provided certificate.

The TLS structures are extended as follows:

enum { gostr341094, gostr34102001 } SignatureAlgorithm;

select (SignatureAlgorithm) { 
  case gostr341094:
    digitally-signed struct { 
      opaque gostr341194_hash[32];
    };
  case gostr34102001:
    digitally-signed struct { 
      opaque gostr341194_hash[32];
    };
} Signature;

CertificateVerify.signature.gost3411_hash = GOSTR3411(handshake_messages)
3.8. Finished

This message is used as described in section 7.4.11 of [TLS1.2].

\[
\text{Finished.verify_data} = \text{PRF}_\text{GOSTR3411} (\text{master_secret}, \text{finished}\_\text{label}, \\
\text{GOSTR3411(handshake_messages)}) [0..11]
\]

4. Compatibility

Some applications use the cipher suites specified herein with [TLS1.0], using features of [TLS1.2], including cipher-suite dependent PRF, Finished and Certificate Verify computations.

5. Security Considerations

It is RECOMMENDED that software applications verify signature values, subject public keys and algorithm parameters to conform to [GOSTR341001], [GOSTR341094] standards prior to their use.

Use of the same key for signature and key derivation is NOT RECOMMENDED.

It is RECOMMENDED for both client and server to verify the private key usage period, if this extension is present in the certificate.

The cipher suites TLS_GOSTR341094_WITH_28147_CNT_IMIT and TLS_GOSTR341001_WITH_28147_CNT_IMIT proposed hereby, have been analyzed by special certification laboratory of Scientific and Technical Centre "ATLAS" in appropriate levels of target_of_evaluation (TOE).

It is RECOMMENDED to subject the implementations of these cipher suites to examination by an authorized agency with approved methods of cryptographic analysis.

6. IANA Considerations

IANA has assigned the following values for GOST 28147-89 mode ciphers definitions:

\[
\text{enum} \{ \\
\text{gostr341094(21), gostr34102001(22)} \\
\} \text{ClientCertificateType};
\]
enum {
    gostr3411(XX)
} HashType;

CipherSuite TLS_GOSTR341094_WITH_28147_CNT_IMIT = \{0x00,0x80\}
CipherSuite TLS_GOSTR341001_WITH_28147_CNT_IMIT = \{0x00,0x81\}
CipherSuite TLS_GOSTR341094_WITH_NULL_GOSTR3411 = \{0x00,0x82\}
CipherSuite TLS_GOSTR341001_WITH_NULL_GOSTR3411 = \{0x00,0x83\}

[IANA please remove] Note: The above numeric definitions for
CipherSuites and ClientCertificateType have not yet been registered.

7. References

7.1. Normative references

[CPALGS] 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

[CPCMS] Leontiev, S. and G. Chudov, "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)",

[CPPK] Leontiev, S. and D. Shefanovski, "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,
May 2006.

[GOST28147] Government Committee of the USSR for Standards,
"Cryptographic Protection for Data Processing System,
Gosudarstvennym Standard of USSR (In Russian)",

[GOST3431004] Council for Standardization, Metrology and Certification
of the Commonwealth of Independence States (EASC), Minsk,
"Information technology. Cryptographic Data Security.
Formation and verification processes of (electronic)
digital signature based on Asymmetric Cryptographic
7.2. Informative references


Appendix A. ASN.1 Modules

Additional ASN.1 modules, referenced here, can be found in [CPALGS] and [CPCMS].

A.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 --
-- The types and values defined in this module are exported for
-- use in the other ASN.1 modules contained within the Russian
-- Cryptography "GOST" & "GOST R" Specifications, and for the use
-- of other applications which will use them to access Russian
-- Cryptography services. Other applications may use them for
-- their own purposes, but this will not constrain extensions and
-- modifications needed to maintain or improve the Russian
-- Cryptography service.
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

Appendix B. Acknowledgments

This document was created in accordance with "Russian Cryptographic Software Compatibility Agreement", signed by FGUE STC "Atlas", CRYPTO-PRO, Factor-TS, MD PREI, Infotecs GmbH, SPRCIS (SPbRCZI), Cryptocom, R-Alph. The aim of this agreement is to achieve mutual compatibility of the products and solutions.

The authors wish to thank:

Microsoft Corporation Russia for provided information about company products and solutions, and also for technical consulting in PKI.
RSA Security Russia and Demos Co Ltd for active collaboration and critical help in creation of this document.
NIP Informzachita for compatibility testing of the proposed data formats while incorporating them into company products.
Citrix Inc for help in compatibility testing Citrix products for Microsoft Windows.
Russ Hously (Vigil Security, LLC, housley@vigilsec.com) and Vasilij Sakharov (DEMOS Co Ltd, svp@dol.ru) for initiative, creating this document.

Author’s Addresses

Alexandr Afanasiev
Factor-TS
office 711, 14, Presnenskij val,
Moscow, 123557, Russian Federation
EMail: afa1@factor-ts.ru

Nikolaj Nikishin
Infotecs GmbH
p/b 35, 80-5, Leningradskij prospekt,
Moscow, 125315, Russian Federation
EMail: nikishin@infotecs.ru

Boleslav Izotov
FGUE STC "Atlas"
38, Obraztsova,
Moscow, 127018, Russian Federation
EMail: izotov@nii.voskhod.ru

Elena Minaeva
MD PREI
build 3, 6A, Vtoroj Troitskij per.,
Moscow, Russian Federation
EMail: evminaeva@mail.ru

Serguei Murugov
R-Alpha
4/1, Raspletina,
Moscow, 123060, Russian Federation
EMail: msm@top-cross.ru

Igor Ustinov
Cryptocom
office 239, 51, Leninskij prospekt,
Moscow, 119991, Russian Federation
EMail: igus@cryptocom.ru

Anatolij Erkin
SPRCIS (SPbRCZI)
1, Obrucheva,
St.Petersburg, 195220, Russian Federation
EMail: erkin@nevsky.net
Authors’ Addresses

Grigorij S. Chudov (editor)
CRYPTO-PRO, Ltd.
38, Obraztsova
Moscow  127018
Russia

Phone: +7 (495) 933 11 68
Fax: +7 (495) 933 11 68
Email: chudov@CryptoPro.ru
URI: http://www.CryptoPro.ru

Serguei E. Leontiev (editor)
CRYPTO-PRO, Ltd.
38, Obraztsova
Moscow  127018
Russia

Phone: +7 (495) 933 11 68
Fax: +7 (495) 933 11 68
Email: lse@CryptoPro.ru
URI: http://www.CryptoPro.ru
Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).