Using GOST R 34.10-2012 and GOST R 34.11-2012 algorithms with the
Internet X.509 Public Key Infrastructure
draft-deremin-rfc4491-bis-00

Abstract

This document updates RFC 3279 and RFC 4491. It describes encoding
formats, identifiers, and parameter formats for the algorithms GOST R
34.10-2012 and GOST R 34.11-2012 for use in Internet X.509 Public Key
Infrastructure (PKI).

Status of This Memo

This Internet-Draft is submitted in full conformance with the
provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering
Task Force (IETF). Note that other groups may also distribute
working documents as Internet-Drafts. The list of current Internet-
Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months
and may be updated, replaced, or obsoleted by other documents at any
time. It is inappropriate to use Internet-Drafts as reference
material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 1, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the
document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Provisions Relating to IETF Documents
(https://trustee.ietf.org/license-info) in effect on the date of
publication of this document. Please review these documents
carefully, as they describe your rights and restrictions with respect
to this document. Code Components extracted from this document must
1. Introduction

This document supplements [RFC3279]. It describes the conventions for using the GOST R 34.10-2012 [GO3410-2012] (see [RFC7091]) signature algorithm and GOST R 34.11-2012 [GO3411-2012] (see...
This specification defines the contents of the signatureAlgorithm, signatureValue, signature, and subjectPublicKeyInfo fields within X.509 Certificates and CRLs. For each algorithm, the appropriate alternatives for the keyUsage certificate extension are provided.

1.1. Requirements Language

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. Signature algorithm support

Conforming CAs MAY use GOST R 34.10-2012 signature algorithm to sign certificates and CRLs. This signature algorithm MUST always be used with GOST R 34.11-2012 hash function. It may use keys length of either 256 bits or 512 bits.

The ASN.1 object identifier used to identify GOST R 34.10-2012 signature algorithm with 256-bit key length and GOST R 34.11-2012 hash function with 256-bit hash code is:

```
id-tc26-signwithdigest-gost3410-12-256 OBJECT IDENTIFIER ::= 
{ iso(1) member-body(2) ru(643) rosstandart(7) tc26(1) 
  algorithms(1) signwithdigest(3) gost3410-12-256(2)}.
```

GOST R 34.10-2012 signature algorithm with 256-bit key length generates a digital signature in the form of two 256-bit numbers, r and s. Its octet string representation consists of 64 octets, where the first 32 octets contain the big-endian representation of s and the second 32 octets contain the big-endian representation of r.

The ASN.1 object identifier used to identify GOST R 34.10-2012 signature algorithm with 512-bit key length and GOST R 34.11-2012 hash function with 512-bit hash code is:

```
id-tc26-signwithdigest-gost3410-12-512 OBJECT IDENTIFIER ::= 
{ iso(1) member-body(2) ru(643) rosstandart(7) tc26(1) 
  algorithms(1) signwithdigest(3) gost3410-12-512(3)}.
```

GOST R 34.10-2012 signature algorithm with 512-bit key length generates a digital signature in the form of two 512-bit numbers, r and s. Its octet string representation consists of 128 octets, where the first 64 octets contain the big-endian representation of s and the second 64 octets contain the big-endian representation of r.
When either of these OID are used as the algorithm field in an AlgorithmIdentifier structure, the encoding MUST omit the parameters field.

The described definition of a signature value is directly usable in CMS [RFC5652], where such values are represented as octet strings. However, signature values in certificates and CRLs [RFC5280] are represented as bit strings, and thus the octet string representation must be converted.

To convert an octet string signature value to a bit string, the most significant bit of the first octet of the signature value SHALL become the first bit of the bit string, and so on through the least significant bit of the last octet of the signature value, which SHALL become the last bit of the bit string.

3. Hash functions support

The ASN.1 object identifier used to identify GOST R 34.11-2012 hash function with 256-bit hash code is:

\[
\text{id-tc26-digest-gost3411-12-256 \hspace{1em} \text{OBJECT IDENTIFIER} \hspace{1em} ::= \hspace{1em} \\
\{ \hspace{1em} \text{iso(1) member-body(2) ru(643) rosstandart(7) tc26(1)} \hspace{1em} \\
\text{algorithms(1) digest(2) gost3411-12-256(2)} \hspace{1em} \}. \\
\]

The ASN.1 object identifier used to identify GOST R 34.11-2012 hash function with 512-bit hash code is:

\[
\text{id-tc26-digest-gost3411-12-512 \hspace{1em} \text{OBJECT IDENTIFIER} \hspace{1em} ::= \hspace{1em} \\
\{ \hspace{1em} \text{iso(1) member-body(2) ru(643) rosstandart(7) tc26(1)} \hspace{1em} \\
\text{algorithms(1) digest(2) gost3411-12-512(3)} \hspace{1em} \}. \\
\]

When either of these OID are used as the algorithm field in an AlgorithmIdentifier structure, the encoding MUST omit the parameters field.

4. Subject Public Keys Information Fields

4.1. Public Key identifiers

GOST R 34.10-2012 public keys with 256 bits private key length are identified by the following OID:

\[
\text{id-tc26-gost3410-12-256 \hspace{1em} \text{OBJECT IDENTIFIER} \hspace{1em} ::= \hspace{1em} \\
\{ \hspace{1em} \text{iso(1) member-body(2) ru(643) rosstandart(7) tc26(1)} \hspace{1em} \\
\text{algorithms(1) sign(1) gost3410-12-256(1)} \hspace{1em} \}. \\
\]
4.2. Public Key parameters

When either of these identifiers appear as algorithm field in SubjectPublicKeyInfo.algorithm.algorithm field, parameters field MUST have the following structure:

GostR3410-2012-PublicKeyParameters ::= SEQUENCE
    {
        publicKeyParamSet OBJECT IDENTIFIER,
        digestParamSet OBJECT IDENTIFIER OPTIONAL
    }

where:

- publicKeyParamSet - public key parameters identifier for GOST R 34.10-2012 (see Sections 5.1 and 5.2 of [RFC7836] or Appendix B) or GOST R 34.10-2001 (see Section 8.4 of [RFC4357]) parameters.

- digestParamSet - parameter identifier for corresponding GOST R 34.11-2012 (See Section 3).

The field digestParamSet:

- SHOULD be omitted if GOST R 34.10-2012 signature algorithm is used with 512-bit key length;

- MUST be present and must be equal to id-tc26-digest-gost3411-12-256 id one of the following values is used as publicKeyParamSet:
  * "id-GostR3410-2001-CryptoPro-A-ParamSet",
  * "id-GostR3410-2001-CryptoPro-B-ParamSet",
  * "id-GostR3410-2001-CryptoPro-C-ParamSet",
  * "id-GostR3410-2001-CryptoPro-XchA-ParamSet",
  * "id-GostR3410-2001-CryptoPro-XchB-ParamSet";
4.3. Public Key encoding

The GOST R 34.10-2012 public key MUST be ASN.1 DER encoded as an OCTET STRING. This encoding SHALL be used as the content (i.e., the value) of the subjectPublicKey field (a BIT STRING) of SubjectPublicKeyInfo structure.

GostR3410-2012-256-PublicKey ::= OCTET STRING (64),
GostR3410-2012-512-PublicKey ::= OCTET STRING (128).

"GostR3410-2012-256-PublicKey" MUST contain 64 octets, where the first 32 octets contain the little-endian representation of "x" and the second 32 octets contains the little-endian representation of "y" coordinates of the public key.

"GostR3410-2012-512-PublicKey" MUST contain 128 octets, where the first 64 octets contain the little-endian representation of "x" and the second 64 octets contains the little-endian representation of "y" coordinates of the public key.

4.4. Key usage extension

If the KeyUsage extension is present in a certificate with GOST R 34.10-2012 public key, the following values MAY be present:

- "digitalSignature" (0),
- "contentCommitment" (1),
- "keyAgreement" (4),
- "keyCertSign" (5),
- "cRLSign" (6),
o "encipherOnly" (7),

o "decipherOnly" (8).

Note that "contentCommitment" was named "nonRepudiation" in previous versions of X.509.

If the key is going to be used for key agreement, flag "keyAgreement" MUST be present in "KeyUsage" extension with "encipherOnly" and "decipherOnly" flags being optional. However flags "encipherOnly" and "decipherOnly" flags MUST NOT be present simultaneously.

5. Qualified certificates extensions

This section defines additional object identifiers (OIDs) for use in qualified certificates for checking digital signatures.

5.1. Distinguished Name additions

OGRN is the main state registration number of juridical entities.

OGRN ::= NUMERIC STRING 13

Corresponding OID is "1.2.643.100.1".

SNILS is the individual insurance account number.

SNILS ::= NUMERIC STRING 11

Corresponding OID is "1.2.643.100.3".

OGRNIP is the main state registration number of individual entrepreneurs.

OGRNIP ::= NUMERIC STRING 15

Corresponding OID is "1.2.643.100.5".

INN is the individual taxpayer number (ITN).

INN ::= NUMERIC STRING 12

Corresponding OID is "1.2.643.3.131.1.1".
5.2. Certificate policies

Russian national regulation body for cryptography defines several security levels of cryptographic tools. Depending on the class of cryptographic token used by certificate owner the following OIDs must be included into certificate policies. Certificate should include OIDs starting from the lowest one (KC1) up to the strongest applicable.

- "1.2.643.100.113.1" - class KC1,
- "1.2.643.100.113.2" - class KC2,
- "1.2.643.100.113.3" - class KC3,
- "1.2.643.100.113.4" - class KB1,
- "1.2.643.100.113.5" - class KB2,
- "1.2.643.100.113.6" - class KA1.

5.3. Subject Sign Tool

To denote the token or software type used by certificate owner following non-critical "SubjectSignTool" extension with OID "1.2.643.100.111" should be included. It is defined as

SubjectSignTool ::= UTF8String SIZE(1..200).

5.4. Issuer Sign Tool

To denote the tools used to generate key pair and tools used by CA to sign certificate following non-critical "IssuerSignTool" extension with OID "1.2.643.100.112" should be included. It is defined as

IssuerSignTool ::= SEQUENCE {
  signTool UTF8String SIZE(1..200),
  cATool UTF8String SIZE(1..200),
  signToolCert UTF8String SIZE(1..100),
  cAToolCert UTF8String SIZE(1..100)},

where:

- "signTool" identifies tools used to create key pair,
- "cATool" identifies tools used by certificate authority,
6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

It is RECOMMENDED that applications verify signature values and subject public keys to conform to [RFC7091] standard prior to their use.

It is RECOMMENDED that CAs and applications make sure that the private key for creating signatures is not used for more than its allowed validity period (typically 15 months for GOST R 34.10-2012 algorithm).

For security discussion concerning use of algorithm parameters, see [ANS17] and the Security Considerations sections in [RFC4357], [RFC7836].

8. References

8.1. Normative References


Informative References


Appendix A. GostR3410-2012-PKISyntax

GostR3410-2012-PKISyntax
{ iso(1) member-body(2) ru(643) rosstandart(7)
tc26(1) modules(0) gostR3411-2012-PKISyntax(2) }

DEFINITIONS ::= BEGIN
    -- EXPORTS All --
-- ASN.1 TC 26 root
id-tc26 OBJECT IDENTIFIER ::= 
    { iso(1) member-body(2) ru(643) rosstandart(7) tc26(1) }

-- Signature algorithm
id-tc26-sign OBJECT IDENTIFIER ::= 
    { id-tc26 algorithms(1) sign(1) }

-- Signature algorithm parameters
id-tc26-sign-constants OBJECT IDENTIFIER ::= 
    { id-tc26 constants(2) sign(1) }

-- GOST R 34.10-2012 / 256 bits signature algorithm parameters
id-tc26-gost-3410-2012-256-constants OBJECT IDENTIFIER ::= 
    { id-tc26-sign-constants gost-3410-2012-256(1) }

-- GOST R 34.10-2012 / 512 bits signature algorithm parameters
id-tc26-gost-3410-2012-512-constants OBJECT IDENTIFIER ::= 
    { id-tc26-sign-constants gost-3410-2012-512(2) }

-- GOST R 34.10-2012 / 256 bits signature algorithm
id-tc26-gost3410-2012-256 OBJECT IDENTIFIER ::= 
    { id-tc26-sign gost3410-2012-256(1) }

-- GOST R 34.10-2012 / 512 bits signature algorithm
id-tc26-gost3410-2012-512 OBJECT IDENTIFIER ::= 
    { id-tc26-sign gost3410-2012-512(2) }

-- Signature & hash algorithm GOST R 34.10-2012 / 256 bits
-- with GOST R 34.11-2012
id-tc26-signwithdigest-gost3410-2012-256 OBJECT IDENTIFIER ::= 
    { id-tc26-signwithdigest gost3410-2012-256(2) }

-- Signature & hash algorithm GOST R 34.10-2012 / 512 bits
-- with GOST R 34.11-2012
id-tc26-signwithdigest-gost3410-2012-512 OBJECT IDENTIFIER ::= 
    { id-tc26-signwithdigest gost3410-2012-512(3) }

-- GOST R 34.10-2012 / 256 bits Signature algorithm parameters ID:
-- "Set A"
id-tc26-gost-3410-2012-256-paramSetA OBJECT IDENTIFIER ::= 
    { id-tc26-gost-3410-2012-256-constants paramSetA(1) }

-- GOST R 34.10-2012 / 256 bits signature algorithm parameters ID:
-- "Set B"
id-tc26-gost-3410-2012-256-paramSetB OBJECT IDENTIFIER ::= 
    { id-tc26-gost-3410-2012-256-constants paramSetB(2) }
-- GOST R 34.10-2012 / 256 bits signature algorithm parameters ID:
-- "Set C"
id-tc26-gost-3410-2012-256-paramSetC OBJECT IDENTIFIER ::= 
{ id-tc26-gost-3410-2012-256-constants paramSetC(3) }

-- GOST R 34.10-2012 / 256 bits signature algorithm parameters ID:
-- "Set D"
id-tc26-gost-3410-2012-256-paramSetD OBJECT IDENTIFIER ::= 
{ id-tc26-gost-3410-2012-256-constants paramSetD(4) }

-- GOST R 34.10-2012 / 512 bits signature algorithm parameters ID:
-- "Test set"
id-tc26-gost-3410-2012-512-paramSetTest OBJECT IDENTIFIER ::= 
{ id-tc26-gost-3410-2012-512-constants paramSetTest(0) }

-- GOST R 34.10-2012 / 512 bits signature algorithm parameters ID:
-- "Set A"
id-tc26-gost-3410-2012-512-paramSetA OBJECT IDENTIFIER ::= 
{ id-tc26-gost-3410-2012-512-constants paramSetA(1) }

-- GOST R 34.10-2012 / 512 bits signature algorithm parameters ID:
-- "Set B"
id-tc26-gost-3410-2012-512-paramSetB OBJECT IDENTIFIER ::= 
{ id-tc26-gost-3410-2012-512-constants paramSetB(2) }

-- GOST R 34.10-2012 / 512 bits signature algorithm parameters ID:
-- "Set C"
id-tc26-gost-3410-2012-512-paramSetC OBJECT IDENTIFIER ::= 
{ id-tc26-gost-3410-2012-512-constants paramSetC(3) }

-- Public key GOST R 34.10-2012 / 256 bits
GostR3410-2012-256-PublicKey ::= OCTET STRING (SIZE (64))

-- Public key GOST R 34.10-2012 / 512 bits
GostR3410-2012-512-PublicKey ::= OCTET STRING (SIZE (128))

-- Public key parameters GOST R 34.10-2012
GostR3410-2012-PublicKeyParameters ::= 
SEQUENCE { 

  publicKeyParamSet OBJECT IDENTIFIER,
  digestParamSet OBJECT IDENTIFIER OPTIONAL
}

END -- GostR3410-2012-PKISyntax
Appendix B. Public key parameters

Here we define three new object identifiers for three existing public key parameter sets defined in [RFC4357]. These object identifiers MUST be used with GOST R 34.10-2012 public keys only.

id-tc26-gost-3410-2012-256-paramSetB OBJECT IDENTIFIER ::= 
{ iso(1) member-body(2) ru(643) rosstandart(7) tc26(1) 
  ru(643) rosstandart(7) tc26(1) constants(2) sign-constants(1) 
gost-3410-12-256-constants(1) paramSetB(2) }.

The elliptic curve of this parameter set is the same as of id-GostR3410-2001-CryptoPro-A-ParamSet which can be found in [RFC4357].

id-tc26-gost-3410-2012-256-paramSetC OBJECT IDENTIFIER ::= 
{ iso(1) member-body(2) ru(643) rosstandart(7) tc26(1) 
  ru(643) rosstandart(7) tc26(1) constants(2) sign-constants(1) 
gost-3410-12-256-constants(1) paramSetC(3) }.

The elliptic curve of this parameter set is the same as of id-GostR3410-2001-CryptoPro-B-ParamSet which can be found in [RFC4357].

id-tc26-gost-3410-2012-256-paramSetD OBJECT IDENTIFIER ::= 
{ iso(1) member-body(2) ru(643) rosstandart(7) tc26(1) 
  ru(643) rosstandart(7) tc26(1) constants(2) sign-constants(1) 
gost-3410-12-256-constants(1) paramSetD(4) }.

The elliptic curve of this parameter set is the same as of id-GostR3410-2001-CryptoPro-C-ParamSet which can be found in [RFC4357].

Appendix C. Test Examples

C.1. GOST R 34.10-2001 Test parameters (256 bit private key length)

This example uses curve defined in Section 7.1 of [RFC7091].

Private key is

d = 0x7A929ADE789BB9BE10ED359DD39A72C11B60961F49397EEE1D19CE9891EC3B28

Public key is

X = 0x7F2B49E270DB6D90D8595BE458B50C58585BA1D4E9B788F6689DBD8E56FD80B
Y = 0x26F1B489D6701DD185C8413A977B3CBBAF64D1C593D266627DFFB101A87FF77DA
C.1.1. Certificate request

-----BEGIN CERTIFICATE REQUEST-----
MIHTM1GBgEAMBixEDAOBgNVBAMTB0V4YWlwbGUwZjAfBgghQMAQEBATATBgcq
hQMCAlMABgggQMHAQECAgNDAARAC9hv5djb1WaPeJtOHBqFhcVQi0XsWlnYkg3b
cOJK3/ad/+HGhD7ydm0pFF0Svuzz71zpByIHRHVDibTxEjQAAMAoGCCqFAwC
AQMCAdEAaqzjXUqqUXi2EVi1efTLwWljz3rMQypBqij8asUgoDN
ntVv7aQ2dAU1VKqNZ7g60EF9odEkw==
-----END CERTIFICATE REQUEST-----

0 211: SEQUENCE {
3 129:   SEQUENCE {
6  1:     INTEGER 0
9  18:     SEQUENCE {
11 16:       SET {
13 14:         SEQUENCE {
15  3:           OBJECT IDENTIFIER commonName (2 5 4 3)
20  7:           PrintableString 'Example'
29 102:     SEQUENCE {
31 31:       SEQUENCE {
33  8:         OBJECT IDENTIFIER '1 2 643 7 1 1 1 1'
43  7:       SEQUENCE {
45  7:         OBJECT IDENTIFIER testSignParams (1 2 643 2 2 35 0)
54  8:         OBJECT IDENTIFIER '1 2 643 7 1 1 2 2'
64  67:       BIT STRING, encapsulates {
67 64:         OCTET STRING
     6A AA B3 8E 35 D4 AA A5 17 94 03 01 79 91 22 D8
     C5 50 8B 45 EC 5B 59 D8 90 6D DB 70 E2 49 2B 7F
     DA 77 FF 87 1A 10 FB DF 27 66 D2 93 C5 D1 64 AF
     BB 3C 7B 97 3A 41 C8 85 D1 1D 70 D6 89 B4 F1 26
     }}}
133  0:     [0] {}
135 10:     SEQUENCE {
137  8:       OBJECT IDENTIFIER '1 2 643 7 1 1 3 2'
147  65:     BIT STRING
    6A AA B3 8E 35 D4 AA A5 17 94 03 01 79 91 22 D8
    55 48 4F 57 9F 4C BB 96 D6 3C DF DF 3A CC 43 2A
    41 AA 28 D2 F1 AB 14 82 80 CD 9E D5 6F ED A4 19
    74 05 35 54 A4 27 67 B8 3A D0 43 FD 39 DC 04 93
    }

C.1.2. Certificate

-----BEGIN CERTIFICATE-----
MIIBGDCBxqADAgECAgEKMAoGCCqFAwcBAQMCMBIxEDAOBgNVBAwMTB0V4YW1wbGUw
IBcMMDExMTAxMDAwMDAwWhgPMjA1MDExMDEwMDEwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
MDaw-----END CERTIFICATE-----

0 280: SEQUENCE {
  4 198:   SEQUENCE {
    7  3:     [0] {
        9  1:       INTEGER 2  
        :         }
  12  1:     INTEGER 10  
  15 10:     SEQUENCE {
    17  8:       OBJECT IDENTIFIER '1 2 643 7 1 1 3 2'  
    :         }
  27 18:     SEQUENCE {
    29 16:     SET {
      31 14:       SEQUENCE {
        33  3:         OBJECT IDENTIFIER commonName (2 5 4 3)  
        38  7:           PrintableString 'Example'  
          :             }
      :         }
  47 32:     SEQUENCE {
    49 13:     UTCTime 01/01/2001 00:00:00 GMT  
    64 15:     GeneralizedTime 31/12/2050 00:00:00 GMT  
          :           }
  81 18:     SEQUENCE {
    83 16:     SET {
      85 14:       SEQUENCE {
        87  3:         OBJECT IDENTIFIER commonName (2 5 4 3)  
        92  7:           PrintableString 'Example'  
          :             }
      :         }
 101 102:    SEQUENCE {
 103 31:     SEQUENCE {
    105  8:       OBJECT IDENTIFIER '1 2 643 7 1 1 1 1'  
    115 19:     SEQUENCE {
      117  7:         OBJECT IDENTIFIER testSignParams (1 2 643 2 2 35 0)  
      126  8:         OBJECT IDENTIFIER '1 2 643 7 1 1 2 2'  
        :             }
      :           }
}
136  67: BIT STRING, encapsulates {
139  64: OCTET STRING
 : 0B D8 6F E5 D8 DB 89 66 8F 78 9B 4E 1D BA 85 85
 : C5 50 8B 45 EC 5B 59 D8 90 6D DB 70 E2 49 2B 7F
 : DA 77 FF 87 1A 10 FB DF 27 66 D2 93 C5 D1 64 AF
 : BB 3C 7B 97 3A 41 C8 85 D1 1D 70 D6 89 B4 F1 26
 : }
 : }
 : }
205  10: SEQUENCE {
207  8: OBJECT IDENTIFIER ’1 2 643 7 1 1 3 2’
 : }
217  65: BIT STRING
 : 43 86 0E 5C 41 40 57 13 3C 48 12 12 CE 90
 : 93 F2 66 AC B3 51 5B 88 3A 24 26 AC C6 0B 6B BE
 : 41 AA 28 D2 F1 AB 14 82 80 CD 9E D5 6F ED A4 19
 : 74 05 35 54 A4 27 67 B8 3A D0 43 FD 39 DC 04 93
 : }

C.1.3. Certificate Revocation List
C.2. GOST R 34.10-2012 TC26-256-A parameters (256 bit private key length)

[[Q1: Provide examples using Edwards curve. --DES]]

C.3. GOST R 34.10-2012 Test parameters (512 bit private key length)

This example uses curve [[Q2: Define the curve or reference a document. --DES]]

Private key is

\[
d = 0x0BA6048AAADAAE241BA40936D47756D7C93091A0E8514669700EE7508E508B1020\\
  72E8123B2200A0563322DAD2827E2714A2636B7BFD18AADFC62967821FA18DD4
\]
Public key is

\[
X = 0x115DC5BC96760C7B48598D8AB9E740D4C4A85A65BE33C181B5C320C854621DD\\
   5A515B56D13314AF69B5C5B92C8B4DDFF75C45415159D9D9D33612CD530EFE1
\]

\[
Y = 0x37C7C90CD40BF5621DC3AC1B751CFA0E2634FA0503B3D52639F5D7FB72AFD61\\
   EA19941D943FFE7F0C70A2759A3CDB84C114E1F9339FDF27F35ECA93677BEEC
\]

C.3.1. Certificate request

-----BEGIN CERTIFICATE REQUEST-----

MIIBTzCBvAIBADASMRAwDgYDVQQDEwdFeGFtcGxlMIGmBCCqAwIBAgIBADASBh

-----END CERTIFICATE REQUEST-----

0 335: SEQUENCE {
4 188:  SEQUENCE {
7  1:    INTEGER 0
10 18:   SEQUENCE {
12 16:    SET {
14 14:      SEQUENCE {
16  3:        OBJECT IDENTIFIER commonName (2 5 4 3)
21  7:          PrintableString 'Example'

30 160:   SEQUENCE {
33 23:    SEQUENCE {
35  8:      OBJECT IDENTIFIER '1 2 643 7 1 1 1 2'
45 11:     SEQUENCE {
47  9:       OBJECT IDENTIFIER '1 2 643 7 2 1 2 0'

58 132:    BIT STRING, encapsulates {
62 128:      OCTET STRING :

C.3.2. Certificate

-----BEGIN CERTIFICATE-----
MIIBlTCCAQGgAwIBAgIBCzAKBggqhQMHAQEDAzASMRAwDgYDVQQDEwdFeGFltcGxlMCAx
DTAxMDEwMDAwMDAwMFoYDzIwNTAxMjMxMDAwMjASMRAwDgYDVQQDEwdFeGFtcGxl
MIIBlTCCAQGgAwIBAgIBCzAKBggqhQMHAQEDAzASMRAwDgYDVQQDEwdFeGFltcGxl
MCAx

-----END CERTIFICATE-----

0 405: SEQUENCE {
  4 257:  SEQUENCE {
    8 3:    [0] {
      10 1:      INTEGER 2
        :    }
    13 1:      INTEGER 11
    16 10:      SEQUENCE {
      18 8:        OBJECT IDENTIFIER ’1 2 643 7 1 1 3 3’
        :    }
  28 18:   SEQUENCE {
    30 16:     SET {
      32 14:       SEQUENCE {
        34 3:         OBJECT IDENTIFIER commonName (2 5 4 3)
        39 7:           PrintableString ’Example’
          :    }
      :    }
  :    }

Eremin-Solenikov, et al. Expires May 1, 2020
48 32:  SEQUENCE {
50 13:    UTCTime 01/01/2001 00:00:00 GMT
65 15:    GeneralizedTime 31/12/2050 00:00:00 GMT
    }
82 18:  SEQUENCE {
84 16:    SET {
86 14:      SEQUENCE {
88  3:        OBJECT IDENTIFIER commonName (2 5 4 3)
93  7:        PrintableString 'Example'
        }
    }
102 160:  SEQUENCE {
105 23:    SEQUENCE {
107  8:      OBJECT IDENTIFIER '1 2 643 7 1 1 1 2'
117 11:    SEQUENCE {
119  9:      OBJECT IDENTIFIER '1 2 643 7 1 2 1 2 0'
        }
    }
130 132:  BIT STRING, encapsulates {
134 128:    OCTET STRING
       :     E1 EF 30 D5 2C 61 33 DD D9 9D 1D 5C 41 45 5C F7
       :      DF 4D 8B 4C 92 5B BC 69 AF 14 33 D1 56 58 51 5A
       :      DD 21 46 85 0C 32 5C 5B 81 C1 33 BE 65 5A A8 C4
       :      D4 40 E7 B9 8A 8D 59 48 7B 0C 76 96 BC C5 5D 11
       :      EC BE 77 36 A9 EC 35 7F F2 FD 39 93 1F 4E 11 4C
       :      B8 CD A3 59 27 0A C7 F0 E7 FF 43 D9 41 94 19 EA
       :      61 FD 2A B7 7F 5D 9F 63 52 3D 5B 50 A0 4F 63 E2
       :      A0 CF 51 B7 C1 3A DC 21 56 0F 0B D4 0C C9 C7 37
       :    }
265 10:  SEQUENCE {
267  8:    OBJECT IDENTIFIER '1 2 643 7 1 1 3 3'
      :  }
277 129:  BIT STRING
       :     3D 55 DE 6D 87 C3 D5 F8 4E E8 46 8E 8E 8B 2D 96
       :     E0 34 12 E7 E7 A7 62 7D A0 83 E7 C5 76 74 26 DB
       :     6A F6 31 53 72 19 15 32 0A 3D A1 F9 A3 46 33 BA
       :     FC ED 0E F6 04 F7 2F 1B E5 2F 37 A2 18 12 92 6C
       :     2F 86 FA 60 A0 81 09 1A 23 DD 79 5E 1E 3C 68 9E
       :     E5 12 A3 C8 2E E0 DC C2 64 3C 78 EE A8 FC AC D3
       :     54 92 55 84 86 B2 0F 1C 9E C1 97 C9 06 99 85 02
       :     60 C9 3B CB CD 9C 5C 33 17 E1 93 44 E1 73 AE 36
       :   }
C.3.3. Certificate Revocation List

-----BEGIN X509 CRL-----
MIHTMEECAQEwCgYIKoUDBwEBAwMwEjEQMA4GA1UEAxMHXhhbXBSZReNMTQwMTAx
MDAwMDAwWhcNMTQwMTAyMDAwMDAwJAKBggqhQMAQEDAwOgYQHw==/t67NtVYO72
E3z8xZGkXMuV7NpCh/Ax+i/k7uoIMH1kjU3AmGxGqHs/vx69C6jQ1nH1ZVMo5/z
q77ZBR9NL4b6YKBCRojX3x485juuUS08gu4NzCZDi47q8rNNUK1WEhrIPHJ7B
18KgYUCYMk7y82cXDMX4ZNE4X0uNg==
-----END X509 CRL-----

0 211: SEQUENCE {
3 65: SEQUENCE {
5 1: INTEGER 1
8 10: SEQUENCE {
10 8: OBJECT IDENTIFIER '1 2 643 7 1 1 3 3'
  : }
20 18: SEQUENCE {
22 16: SET {
24 14: SEQUENCE {
26 3: OBJECT IDENTIFIER commonName (2 5 4 3)
31 7: PrintableString 'Example'
  : }
  : }
40 13: UTCTime 01/01/2014 00:00:00 GMT
55 13: UTCTime 02/01/2014 00:00:00 GMT
  : }
70 10: SEQUENCE {
72 8: OBJECT IDENTIFIER '1 2 643 7 1 1 3 3'
  : }
82 129: BIT STRING
  : 3A 13 FB 7A EC DB 55 60 EE F6 13 7C FC 5D D6 46
  : 91 73 2E BF B3 69 0A 1F C0 C7 E8 A4 EE EA 08 30
  : 7D 64 8D 4D C0 98 6C 46 A8 7B 3F BE 4C 7A F4 2E
  : A3 43 59 C7 95 95 4C A3 9F F3 AB BE D9 05 1F 4D
  : 2F 86 FA 60 A0 81 09 1A 23 DD 79 5E 1E 3C 68 9E
  : E5 12 A3 C8 2E E0 DC C2 64 3C 78 EE A8 FC AC D3
  : 54 92 55 84 86 B2 0F 1C 9E C1 97 C9 06 99 85 02
  : 60 C9 3B CB CD 9C 5C 33 17 E1 93 44 E1 73 AE 36
  : }

Appendix D. Contributors

- Semen Pianov
  InfoTeCS JSC
  Semen.Pianov@infotecs.ru

Eremin-Solenikov, et al. Expires May 1, 2020
Authors’ Addresses

Dmitry Eremin-Solenikov (editor)
Mentor Graphics (Ireland) Ltd.
Pevchesky lane, 12
Saint-Petersburg 197046
Russian Federation

Email: dbaryshkov@gmail.com

Vasily Nikolaev
CryptoPro
18, Suschevsky val
Moscow 127018
Russian Federation

Phone: +7 (495) 995-48-20
Email: nikolaev@cryptopro.ru

Aleksandr Chelpanov
InfoTeCS JSC
Bldg. 1, 1/23, Stary Petrovsko-Razumovskiy Proezd
Moscow 127287
Russian Federation

Phone: +7 (495) 737-61-92
Email: Aleksandr.Chelpanov@infotecs.ru