Network Working Group
Request for Comments: 4491
Updates: 3279
Category: Standards Track

S. Leontiev, Ed.
CRYPTO-PRO
D. Shefanovski, Ed.
Mobile TeleSystems OJSC

May 2006

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

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document supplements RFC 3279. It describes encoding formats, identifiers, and parameter formats for the algorithms GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 for use in Internet X.509 Public Key Infrastructure (PKI).

Table of Contents

1. Introduction

This document supplements RFC 3279 [PKALGS]. It describes the conventions for using the GOST R 34.10-94 [GOST3431095, GOSTR341094] and GOST R 34.10-2001 [GOST3431004, GOSTR341001] signature algorithms, VKO GOST R 34.10-94 and VKO GOST R 34.10-2001 key derivation algorithms, and GOST R 34.11-94 [GOST3431195, GOSTR341194] one-way hash function in the Internet X.509 Public Key Infrastructure (PKI) [PROFILE].

This document provides supplemental information and specifications needed by the "Russian Cryptographic Software Compatibility Agreement" community.

The algorithm identifiers and associated parameters are specified for subject public keys that employ the GOST R 34.10-94 [GOSTR341094]/VKO GOST R 34.10-94 [CPALGS] or the GOST R 34.10-2001 [GOSTR341001]/VKO GOST R 34.10-2001 [CPALGS] algorithms, as is the encoding format for the signatures produced by these algorithms. Also, the algorithm identifiers for using the GOST R 34.11-94 one-way hash function with the GOST R 34.10-94 and GOST R 34.10-2001 signature algorithms are specified.

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.

ASN.1 modules, including all the definitions used in this document, can be found in [CPALGS].

1.1. Requirement Words

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. Algorithm Support

This section is an overview of cryptographic algorithms that may be used within the Internet X.509 certificates and CRL profile [PROFILE]. It describes one-way hash functions and digital signature algorithms that may be used to sign certificates and CRLs, and it identifies object identifiers (OIDs) and ASN.1 encoding for public keys contained in a certificate.

Certification authorities (CAs) and/or applications conforming to this standard MUST support at least one of the specified public key and signature algorithms.

2.1. One-Way Hash Function

This section describes the use of a one-way, collision-free hash function GOST R 34.11-94, the only one that can be used in the digital signature algorithm GOST R 34.10-94/2001. The data that is hashed for certificates and CRL signing is fully described in RFC 3280 [PROFILE].

2.1.1. One-Way Hash Function GOST R 34.11-94

GOST R 34.11-94 has been developed by "GUBS of Federal Agency Government Communication and Information" and "All-Russian Scientific and Research Institute of Standardization". The algorithm GOST R 34.11-94 produces a 256-bit hash value of an arbitrary finite bit length input. This document does not contain the full GOST R 34.11-94 specification, which can be found in [GOSTR341194] (in Russian). [Schneier95], ch. 18.11, p. 454, contains a brief technical description in English.

This function MUST always be used with parameter set identified by id-GostR3411-94-CryptoProParamSet (see Section 8.2 of [CPALGS]).

2.2. Signature Algorithms

Conforming CAs may use GOST R 34.10-94 or GOST R 34.10-2001 signature algorithms to sign certificates and CRLs.

These signature algorithms MUST always be used with a one-way hash function GOST R 34.11-94 as indicated in [GOSTR341094] and [GOSTR341001].

This section defines algorithm identifiers and parameters to be used in the signatureAlgorithm field in a Certificate or CertificateList.

2.2.1. Signature Algorithm GOST R 34.10-94

GOST R 34.10-94 has been developed by "GUBS of Federal Agency Government Communication and Information" and "All-Russian Scientific and Research Institute of Standardization". This document does not contain the full GOST R 34.10-94 specification, which can be found in [GOSTR341094] (in Russian). [Schneier95], ch. 20.3, p. 495, contains a brief technical description in English.

The ASN.1 object identifier used to identify this signature algorithm is:

```
id-GostR3411-94-with-GostR3410-94 OBJECT IDENTIFIER ::=
      { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
        gostR3411-94-with-gostR3410-94(4) }
```

When the id-GostR3411-94-with-GostR3410-94 algorithm identifier appears as the algorithm field in an AlgorithmIdentifier, the encoding SHALL omit the parameters field. That is, the AlgorithmIdentifier SHALL be a SEQUENCE of one component: the OBJECT IDENTIFIER id-GostR3411-94-with-GostR3410-94.

The signature algorithm GOST R 34.10-94 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’.

This definition of a signature value is directly usable in CMS [CMS], where such values are represented as octet strings. However, signature values in certificates and CRLs [PROFILE] 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.

2.2.2. Signature Algorithm GOST R 34.10-2001

GOST R 34.10-2001 was developed by "GUBS of Federal Agency Government Communication and Information" and "All-Russian Scientific and Research Institute of Standardization". This document does not contain the full GOST R 34.10-2001 specification, which can be found in [GOSTR341001] (in Russian).

The ASN.1 object identifier used to identify this signature algorithm is:

```
id-GostR3411-94-with-GostR3410-2001 OBJECT IDENTIFIER ::=
      { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
        gostR3411-94-with-gostR3410-2001(3) }
```

When the id-GostR3411-94-with-GostR3410-2001 algorithm identifier appears as the algorithm field in an AlgorithmIdentifier, the encoding SHALL omit the parameters field. That is, the AlgorithmIdentifier SHALL be a SEQUENCE of one component: the OBJECT IDENTIFIER id-GostR3411-94-with-GostR3410-2001.

The signature algorithm GOST R 34.10-2001 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 process described above (Section 2.2.1) MUST be used to convert this octet string representation to a bit string for use in certificates and CRLs.

2.3. Subject Public Key Algorithms

This section defines OIDs and public key parameters for public keys that employ the GOST R 34.10-94 [GOSTR341094]/VKO GOST R 34.10-94 [CPALGS] or the GOST R 34.10-2001 [GOSTR341001]/VKO GOST R 34.10-2001 [CPALGS] algorithms.

Use of the same key for both signature and key derivation is NOT RECOMMENDED. The intended application for the key MAY be indicated in the keyUsage certificate extension (see [PROFILE], Section 4.2.1.3).

2.3.1. GOST R 34.10-94 Keys

GOST R 34.10-94 public keys can be used for the signature algorithm GOST R 34.10-94 [GOSTR341094] and for the key derivation algorithm VKO GOST R 34.10-94 [CPALGS].

GOST R 34.10-94 public keys are identified by the following OID:

```
id-GostR3410-94 OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
        gostR3410-94(20) }
```

The SubjectPublicKeyInfo.algorithm.algorithm field (see RFC 3280 [PROFILE]) for GOST R 34.10-94 keys MUST be set to id-GostR3410-94.

When the id-GostR3410-94 algorithm identifier appears as the algorithm field in an AlgorithmIdentifier, the encoding MAY omit the parameters field or set it to NULL. Otherwise, this field MUST have the following structure:

```
 GostR3410-94-PublicKeyParameters ::=
      SEQUENCE {
          publicKeyParamSet
              OBJECT IDENTIFIER,
          digestParamSet
              OBJECT IDENTIFIER,
          encryptionParamSet
              OBJECT IDENTIFIER DEFAULT
                  id-Gost28147-89-CryptoPro-A-ParamSet
      }
```

where:

* publicKeyParamSet - public key parameters identifier for GOST R 34.10-94 (see Section 8.3 of [CPALGS])
* digestParamSet - parameters identifier for GOST R 34.11-94 (see Section 8.2 of [CPALGS])
* encryptionParamSet - parameters identifier for GOST 28147-89 [GOST28147] (see Section 8.1 of [CPALGS])

The absence of parameters SHALL be processed as described in RFC 3280 [PROFILE], Section 6.1; that is, parameters are inherited from the issuer certificate. When the working_public_key_parameters variable is set to null, the certificate and any signature verifiable on this certificate SHALL be rejected.

The GOST R 34.10-94 public key MUST be ASN.1 DER encoded as an OCTET STRING; this encoding shall be used as the contents (i.e., the value) of the subjectPublicKey component (a BIT STRING) of the SubjectPublicKeyInfo data element.

```
GostR3410-94-PublicKey ::= OCTET STRING -- public key, Y
```

GostR3410-94-PublicKey MUST contain 128 octets of the little-endian representation of the public key Y = a^x (mod p), where a and p are public key parameters, and x is a private key.

Some erroneous applications discard zero bits at the end of BIT STRING containing the public key. It is RECOMMENDED to pad the bit string with zeroes up to 1048 bits (131 octets) on decoding to be able to decode the encapsulated OCTET STRING.

If the keyUsage extension is present in an end-entity certificate that contains a GOST R 34.10-94 public key, the following values MAY be present:

digitalSignature;
nonRepudiation;
keyEncipherment; and
keyAgreement.

If the keyAgreement or keyEnchiperment extension is present in a certificate GOST R 34.10-94 public key, the following values MAY be present as well:

encipherOnly; and
decipherOnly.

The keyUsage extension MUST NOT assert both encipherOnly and decipherOnly.

If the keyUsage extension is present in an CA or CRL signer certificate that contains a GOST R 34.10-94 public key, the following values MAY be present:

digitalSignature;
nonRepudiation;
keyCertSign; and
cRLSign.

2.3.2. GOST R 34.10-2001 Keys

GOST R 34.10-2001 public keys can be used for the signature algorithm GOST R 34.10-2001 [GOSTR341001] and for the key derivation algorithm VKO GOST R 34.10-2001 [CPALGS].

GOST R 34.10-2001 public keys are identified by the following OID:

```
id-GostR3410-2001 OBJECT IDENTIFIER ::=
    { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
        gostR3410-2001(19) }
```

The SubjectPublicKeyInfo.algorithm.algorithm field (see RFC 3280 [PROFILE]) for GOST R 34.10-2001 keys MUST be set to id-GostR3410-2001.

When the id-GostR3410-2001 algorithm identifier appears as the algorithm field in an AlgorithmIdentifier, the encoding MAY omit the parameters field or set it to NULL. Otherwise, this field MUST have the following structure:

```
 GostR3410-2001-PublicKeyParameters ::=
     SEQUENCE {
         publicKeyParamSet
             OBJECT IDENTIFIER,
         digestParamSet
             OBJECT IDENTIFIER,
         encryptionParamSet
             OBJECT IDENTIFIER DEFAULT
                 id-Gost28147-89-CryptoPro-A-ParamSet
     }
```

where:

* publicKeyParamSet - public key parameters identifier for GOST R 34.10-2001 (see Section 8.4 of [CPALGS])
* digestParamSet - parameters identifier for GOST R 34.11-94 (see Section 8.2 of [CPALGS])
* encryptionParamSet - parameters identifier for GOST 28147-89 [GOST28147] (see Section 8.1 of [CPALGS])

The absence of parameters SHALL be processed as described in RFC 3280 [PROFILE], Section 6.1; that is, parameters are inherited from the issuer certificate. When the working_public_key_parameters variable is set to null, the certificate and any signature verifiable on this certificate SHALL be rejected.

The GOST R 34.10-2001 public key MUST be ASN.1 DER encoded as an OCTET STRING; this encoding shall be used as the contents (i.e., the value) of the subjectPublicKey component (a BIT STRING) of the SubjectPublicKeyInfo data element.

```
GostR3410-2001-PublicKey ::= OCTET STRING -- public key vector, Q
```

According to [GOSTR341001], a public key is a point on the elliptic curve Q = (x,y).

GostR3410-2001-PublicKey MUST contain 64 octets, where the first 32 octets contain the little-endian representation of x and the second 32 octets contain the little-endian representation of y. This corresponds to the binary representation of (<y>256||<x>256) from [GOSTR341001], ch. 5.3.

Some erroneous applications discard zero bits at the end of BIT STRING containing the public key. It is RECOMMENDED to pad the bit string with zeroes up to 528 bits (66 octets) on decoding to be able to decode the encapsulated OCTET STRING.

The same keyUsage constraints apply for use of GOST R 34.10-2001 keys as described in Section 2.3.1 for GOST R 34.10-94 keys.

3. Security Considerations

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

When a certificate is used to support digital signatures as an analogue to manual ("wet") signatures, in the context of Russian Federal Electronic Digital Signature Law [RFEDSL], the certificate MUST contain keyUsage extension, it MUST be critical, and keyUsage MUST NOT include keyEncipherment and keyAgreement.

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 both the GOST R 34.10-94 and GOST R 34.10-2001 algorithms).

For security discussion concerning use of algorithm parameters, see the Security Considerations section in [CPALGS].

4. Examples

4.1. GOST R 34.10-94 Certificate


```
-----BEGIN CERTIFICATE-----
MIICCzCCAboCECMO42BGlSTOxwvklBgufuswCAYGKoUDAgIEMGkxHTAbBgNVBAMM
FEdvc3RSMzQxMC05NCBleGFtcGxlMRIwEAYDVQQKDAlDcnlwdG9Qcm8xCzAJBgNV
BAYTAlJVMScwJQYJKoZIhvcNAQkBFhhHb3N0UjM0MTAtOTRAZXhhbXBsZS5jb20w
HhcNMDUwODE2MTIzMjUwWhcNMTUwODE2MTIzMjUwWjBpMR0wGwYDVQQDDBRHb3N0
UjM0MTAtOTQgZXhhbXBsZTESMBAGA1UECgwJQ3J5cHRvUHJvMQswCQYDVQQGEwJS
VTEnMCUGCSqGSIb3DQEJARYYR29zdFIzNDEwLTk0QGV4YW1wbGUuY29tMIGlMBwG
BiqFAwICFDASBgcqhQMCAiACBgcqhQMCAh4BA4GEAASBgLuEZuF5nls02CyAfxOo
GWZxV/6MVCUhR28wCyd3RpjG+0dVvrey85NsObVCNyaE4g0QiiQOHwxCTSs7ESuo
v2Y5MlyUi8Go/htjEvYJJYfMdRv05YmKCYJo01x3pg+2kBATjeM+fJyR1qwNCCw+
eMG1wra3Gqgqi0WBkzIydvp7MAgGBiqFAwICBANBABHHCH4S3ALxAiMpR3aPRyqB
g1DjB8zy5DEjiULIc+HeIveF81W9lOxGkZxnrFjXBSqnjLeFKgF1hffXOAP7zUM=
-----END CERTIFICATE-----
```


```
  0 30  523: SEQUENCE {
  4 30  442:  SEQUENCE {
  8 02   16:   INTEGER
           :    23 0E E3 60 46 95 24 CE C7 0B E4 94 18 2E 7E EB
 26 30    8:   SEQUENCE {
 28 06    6:    OBJECT IDENTIFIER
           :     id-GostR3411-94-with-GostR3410-94 (1 2 643 2 2 4)
           :    }
 36 30  105:   SEQUENCE {
 38 31   29:    SET {
 40 30   27:     SEQUENCE {
 42 06    3:      OBJECT IDENTIFIER commonName (2 5 4 3)
 47 0C   20:      UTF8String 'GostR3410-94 example'
           :      }
           :     }
 69 31   18:    SET {
 71 30   16:     SEQUENCE {
 73 06    3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
 78 0C    9:      UTF8String 'CryptoPro'
           :      }
           :     }
 89 31   11:    SET {
 91 30    9:     SEQUENCE {
 93 06    3:      OBJECT IDENTIFIER countryName (2 5 4 6)
 98 13    2:      PrintableString 'RU'
           :      }
           :     }
102 31   39:    SET {
104 30   37:     SEQUENCE {
106 06    9:      OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
```

```
 117 16   24:       IA5String 'GostR3410-94@example.com'
            :       }
            :      }
            :     }
 143 30   30:   SEQUENCE {
 145 17   13:    UTCTime '050816123250Z'
 160 17   13:    UTCTime '150816123250Z'
            :    }
 175 30  105:   SEQUENCE {
 177 31   29:    SET {
 179 30   27:     SEQUENCE {
 181 06    3:      OBJECT IDENTIFIER commonName (2 5 4 3)
 186 0C   20:      UTF8String 'GostR3410-94 example'
            :      }
            :     }
 208 31   18:    SET {
 210 30   16:     SEQUENCE {
 212 06    3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
 217 0C    9:      UTF8String 'CryptoPro'
            :      }
            :     }
 228 31   11:    SET {
 230 30    9:     SEQUENCE {
 232 06    3:      OBJECT IDENTIFIER countryName (2 5 4 6)
 237 13    2:      PrintableString 'RU'
            :      }
            :     }
 241 31   39:    SET {
 243 30   37:     SEQUENCE {
 245 06    9:      OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
 256 16   24:      IA5String 'GostR3410-94@example.com'
            :      }
            :     }
            :    }
 282 30  165:   SEQUENCE {
 285 30   28:    SEQUENCE {
 287 06    6:     OBJECT IDENTIFIER id-GostR3410-94 (1 2 643 2 2 20)
 295 30   18:     SEQUENCE {
 297 06    7:      OBJECT IDENTIFIER
            :       id-GostR3410-94-CryptoPro-A-ParamSet
            :        (1 2 643 2 2 32 2)
 306 06    7:      OBJECT IDENTIFIER
            :       id-GostR3411-94-CryptoProParamSet
            :        (1 2 643 2 2 30 1)
            :      }
            :     }
 315 03  132:    BIT STRING 0 unused bits, encapsulates {
 319 04  128:     OCTET STRING
```

```
              :      BB 84 66 E1 79 9E 5B 34 D8 2C 80 7F 13 A8 19 66
              :      71 57 FE 8C 54 25 21 47 6F 30 0B 27 77 46 98 C6
              :      FB 47 55 BE B7 B2 F3 93 6C 39 B5 42 37 26 84 E2
              :      0D 10 8A 24 0E 1F 0C 42 4D 2B 3B 11 2B A8 BF 66
              :      39 32 5C 94 8B C1 A8 FE 1B 63 12 F6 09 25 87 CC
              :      75 1B F4 E5 89 8A 09 82 68 D3 5C 77 A6 0F B6 90
              :      10 13 8D E3 3E 7C 9C 91 D6 AC 0D 08 2C 3E 78 C1
              :      B5 C2 B6 B7 1A A8 2A 8B 45 81 93 32 32 76 FA 7B
              :     }
              :    }
              :   }
 450 30    8:  SEQUENCE {
 452 06    6:   OBJECT IDENTIFIER
              :    id-GostR3411-94-with-GostR3410-94 (1 2 643 2 2 4)
              :   }
 460 03   65:  BIT STRING 0 unused bits
              :   11 C7 08 7E 12 DC 02 F1 02 23 29 47 76 8F 47 2A
              :   81 83 50 E3 07 CC F2 E4 31 23 89 42 C8 73 E1 DE
              :   22 F7 85 F3 55 BD 94 EC 46 91 9C 67 AC 58 D7 05
              :   2A A7 8C B7 85 2A 01 75 85 F7 D7 38 03 FB CD 43
              :  }
```

In the signature of the above certificate, r’ equals
0x22F785F355BD94EC46919C67AC58D7052AA78CB7852A017585F7D73803FBCD43
and s equals
0x11C7087E12DC02F102232947768F472A818350E307CCF2E431238942C873E1DE

4.2. GOST R 34.10-2001 Certificate

-----BEGIN CERTIFICATE-----
MIIB0DCCAX8CECv1xh7CEb0Xx9zUYma0LiEwCAYGKoUDAgIDMG0xHzAdBgNVBAMM
Fkdvc3RSMzQxMC0yMDAxIGV4YW1wbGUxEjAQBgNVBAoMCUNyeXB0b1BybzELMAkG
A1UEBhMCUlUxKTAnBgkqhkiG9w0BCQEWGkdvc3RSMzQxMC0yMDAxQGV4YW1wbGUu
Y29tMB4XDTA1MDgxNjE0MTgyMFoXDTE1MDgxNjE0MTgyMFowbTEfMB0GA1UEAwwW
R29zdFIzNDEwLTIwMDEgZXhhbXBsZTESMBAGA1UECgwJQ3J5cHRvUHJvMQswCQYD
VQQGEwJSVTEpMCcGCSqGSIb3DQEJARYaR29zdFIzNDEwLTIwMDFAZXhhbXBsZS5j
b20wYzAcBgYqhQMCAhMwEgYHKoUDAgIkAAYHKoUDAgIeAQNDAARAhJVodWACGkB1
CM0TjDGJLP3lBQN6Q1z0bSsP508yfleP68wWuZWIA9CafIWuD+SN6qa7flbHy7Df
D2a8yuoaYDAIBgYqhQMCAgMDQQA8L8kJRLcnqeyn1en7U23Sw6pkfEQu3u0xFkVP
vFQ/3cHeF26NG+xxtZPz3TaTVXdoiYkXYiD02rEx1bUcM97i
-----END CERTIFICATE-----

```
   0 30  464: SEQUENCE {
   4 30  383:  SEQUENCE {
   8 02   16:   INTEGER
              :    2B F5 C6 1E C2 11 BD 17 C7 DC D4 62 66 B4 2E 21
  26 30    8:   SEQUENCE {
  28 06    6:    OBJECT IDENTIFIER
```

```
             :     id-GostR3411-94-with-GostR3410-2001 (1 2 643 2 2 3)
             :    }
  36 30  109:   SEQUENCE {
  38 31   31:    SET {
  40 30   29:     SEQUENCE {
  42 06    3:      OBJECT IDENTIFIER commonName (2 5 4 3)
  47 0C   22:      UTF8String 'GostR3410-2001 example'
             :      }
             :     }
  71 31   18:    SET {
  73 30   16:     SEQUENCE {
  75 06    3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
  80 0C    9:      UTF8String 'CryptoPro'
             :      }
             :     }
  91 31   11:    SET {
  93 30    9:     SEQUENCE {
  95 06    3:      OBJECT IDENTIFIER countryName (2 5 4 6)
 100 13    2:      PrintableString 'RU'
             :      }
             :     }
 104 31   41:    SET {
 106 30   39:     SEQUENCE {
 108 06    9:      OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
 119 16   26:      IA5String 'GostR3410-2001@example.com'
             :      }
             :     }
             :    }
 147 30   30:   SEQUENCE {
 149 17   13:    UTCTime '050816141820Z'
 164 17   13:    UTCTime '150816141820Z'
             :    }
 179 30  109:   SEQUENCE {
 181 31   31:    SET {
 183 30   29:     SEQUENCE {
 185 06    3:      OBJECT IDENTIFIER commonName (2 5 4 3)
 190 0C   22:      UTF8String 'GostR3410-2001 example'
             :      }
             :     }
 214 31   18:    SET {
 216 30   16:     SEQUENCE {
 218 06    3:      OBJECT IDENTIFIER organizationName (2 5 4 10)
 223 0C    9:      UTF8String 'CryptoPro'
             :      }
             :     }
 234 31   11:    SET {
 236 30    9:     SEQUENCE {
 238 06    3:      OBJECT IDENTIFIER countryName (2 5 4 6)
```

```
 243 13    2:       PrintableString 'RU'
           :       }
           :      }
 247 31   41:     SET {
 249 30   39:      SEQUENCE {
 251 06    9:       OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
 262 16   26:       IA5String 'GostR3410-2001@example.com'
           :       }
           :      }
           :     }
 290 30   99:    SEQUENCE {
 292 30   28:     SEQUENCE {
 294 06    6:      OBJECT IDENTIFIER id-GostR3410-2001 (1 2 643 2 2 19)
 302 30   18:      SEQUENCE {
 304 06    7:       OBJECT IDENTIFIER
           :        id-GostR3410-2001-CryptoPro-XchA-ParamSet
           :         (1 2 643 2 2 36 0)
 313 06    7:       OBJECT IDENTIFIER
           :        id-GostR3411-94-CryptoProParamSet
           :         (1 2 643 2 2 30 1)
           :       }
           :      }
 322 03   67:     BIT STRING 0 unused bits, encapsulates {
 325 04   64:      OCTET STRING
           :       84 95 68 75 60 02 1A 40 75 08 CD 13 8C 31 89 2C
           :       FD E5 05 03 7A 43 5C F4 6D 2B 0F E7 4F 32 7E 57
           :       8F EB CC 16 B9 95 88 03 D0 9A 7C 85 AE 0F E4 8D
           :       EA A6 BB 7E 56 C7 CB B0 DF 0F 66 BC CA EA 1A 60
           :      }
           :     }
           :    }
 391 30    8:   SEQUENCE {
 393 06    6:    OBJECT IDENTIFIER
           :     id-GostR3411-94-with-GostR3410-2001 (1 2 643 2 2 3)
           :    }
 401 03   65:   BIT STRING 0 unused bits
           :    3C 2F C9 09 44 B7 27 A9 EC A7 D5 E9 FB 53 6D D2
           :    C3 AA 64 7C 44 2E DE ED 31 16 45 4F BC 54 3F DD
           :    C1 DE 17 6E 8D 1B EC 71 B5 93 F3 DD 36 93 55 77
           :    68 89 89 17 62 20 F4 DA B1 31 D5 B5 1C 33 DE E2
           :   }
```

In the public key of the above certificate, x equals
0x577E324FE70F2B6DF45C437A0305E5FD2C89318C13CD0875401A026075689584
and y equals
0x601AEACABC660FDFB0CBC7567EBBA6EA8DE40FAE857C9AD0038895B916CCEB8F
The corresponding private key d equals
0x0B293BE050D0082BDAE785631A6BAB68F35B42786D6DDA56AFAF169891040F77

In the signature of the above certificate, r equals
0xC1DE176E8D1BEC71B593F3DD36935577688989176220F4DAB131D5B51C33DEE2
and s equals
0x3C2FC90944B727A9ECA7D5E9FB536DD2C3AA647C442EDEED3116454FBC543FDD

5. Acknowledgements

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-Alpha. The goal of this agreement is to achieve mutual compatibility of the products and solutions.

The authors wish to thank the following:

Microsoft Corporation Russia for providing 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.

RSA Security Inc for compatibility testing of the proposed data formats while incorporating them into the RSA Keon product.

Baltimore Technology plc for compatibility testing of the proposed data formats while incorporating them into their UniCERT product.

Peter Gutmann for his helpful "dumpasn1" program.

Russ Housley (Vigil Security, LLC, housley@vigilsec.com) and Vasilij Sakharov (DEMOS Co Ltd, svp@dol.ru) for encouraging the authors to create this document.

Grigorij Chudov for navigating the IETF process for this document.

Prikhodko Dmitriy (VSTU, PrikhodkoDV@volgablob.ru) for invaluable assistance in proofreading this document and verifying the form and the contents of the ASN.1 structures mentioned or used in this document.

6. References

6.1. Normative References

[GOST28147] "Cryptographic Protection for Data Processing System", GOST 28147-89, Gosudarstvennyi Standard of USSR, Government Committee of the USSR for Standards, 1989. (In Russian)

[GOST3431195] "Information technology. Cryptographic Data Security. Cashing function.", GOST 34.311-95, Council for Standardization, Metrology and Certification of the Commonwealth of Independence States (EASC), Minsk, 1995. (In Russian)

[GOST3431095] "Information technology. Cryptographic Data Security. Produce and check procedures of Electronic Digital Signature based on Asymmetric Cryptographic Algorithm.", GOST 34.310-95, Council for Standardization, Metrology and Certification of the Commonwealth of Independence States (EASC), Minsk, 1995. (In Russian)

[GOST3431004] "Information technology. Cryptographic Data Security. Formation and verification processes of (electronic) digital signature based on Asymmetric Cryptographic Algorithm.", GOST 34.310-2004, Council for Standardization, Metrology and Certification of the Commonwealth of Independence States (EASC), Minsk, 2004. (In Russian)

[GOSTR341094] "Information technology. Cryptographic Data Security. Produce and check procedures of Electronic Digital Signatures based on Asymmetric Cryptographic Algorithm.", GOST R 34.10-94, Gosudarstvennyi Standard of Russian Federation, Government Committee of the Russia for Standards, 1994. (In Russian)

[GOSTR341001] "Information technology. Cryptographic data security. Signature and verification processes of [electronic] digital signature.", GOST R 34.10-2001, Gosudarstvennyi Standard of Russian Federation, Government Committee of the Russia for Standards, 2001. (In Russian)

[GOSTR341194] "Information technology. Cryptographic Data Security. Hashing function.", GOST R 34.10-94, Gosudarstvennyi Standard of Russian Federation, Government Committee of the Russia for Standards, 1994. (In Russian)

[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 Algorithms", RFC 4357, January 2006.

[PKALGS] Bassham, L., Polk, W., and R. Housley, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3279, April 2002.

[PROFILE] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[X.660] ITU-T Recommendation X.660 Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6.2. Informative References

[Schneier95] B. Schneier, Applied Cryptography, Second Edition, John Wiley & Sons, Inc., 1995.

[RFEDSL] Russian Federal Electronic Digital Signature Law, 10 Jan 2002 N 1-FZ.

[CMS] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004.

Authors' Addresses

Serguei Leontiev, Ed.
CRYPTO-PRO
38, Obraztsova,
Moscow, 127018, Russian Federation

EMail: lse@cryptopro.ru

Dennis Shefanovski, Ed.
Mobile TeleSystems OJSC
4, Marksistskaya Str.,
Moscow, 109147, Russian Federation

EMail: dbs@mts.ru

Grigorij Chudov
CRYPTO-PRO
38, Obraztsova,
Moscow, 127018, Russian Federation

EMail: chudov@cryptopro.ru

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

Igor Ovcharenko
MD PREI
Office 600, 14, B.Novodmitrovskaya,
Moscow, Russian Federation

EMail: igori@mo.msk.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

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.

Acknowledgement

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