

PKIX Working Group  
Internet Draft  
Expires March 8, 2006  
Intended Category: Informational

Serguei Leontiev, CRYPTO-PRO  
Dennis Shefanovski, DEMOS Co Ltd  
September 8, 2005

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.

<[draft-ietf-pkix-gost-cppk-03.txt](#)>

#### Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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 a "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/lid-abstracts.html>.

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on March 8, 2006.

#### Copyright Notice

Copyright (C) The Internet Society (2005).

#### 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).

Internet-Draft

Using GOST with PKIX

September 2005

## Table of Contents

<a href="#">1</a>	Introduction. . . . .	<a href="#">2</a>
<a href="#">2</a>	Algorithm Support . . . . .	<a href="#">3</a>
<a href="#">2.1</a>	One-way Hash Function . . . . .	<a href="#">3</a>
<a href="#">2.1.1</a>	One-way Hash Function GOST R 34.11-94 . . . . .	<a href="#">3</a>
<a href="#">2.2</a>	Signature Algorithms. . . . .	<a href="#">3</a>
<a href="#">2.2.1</a>	Signature Algorithm GOST R 34.10-94 . . . . .	<a href="#">4</a>
<a href="#">2.2.2</a>	Signature Algorithm GOST R 34.10-2001 . . . . .	<a href="#">5</a>
<a href="#">2.3</a>	Subject Public Key Algorithms . . . . .	<a href="#">5</a>
<a href="#">2.3.1</a>	GOST R 34.10-94 Keys. . . . .	<a href="#">6</a>
<a href="#">2.3.2</a>	GOST R 34.10-2001 Keys. . . . .	<a href="#">7</a>
<a href="#">3</a>	Security Considerations . . . . .	<a href="#">9</a>
<a href="#">4</a>	<a href="#">Appendix Examples</a> . . . . .	<a href="#">10</a>
<a href="#">4.1</a>	GOST R 34.10-94 Certificate . . . . .	<a href="#">10</a>
<a href="#">4.2</a>	GOST R 34.10-2001 Certificate . . . . .	<a href="#">12</a>
<a href="#">5</a>	References. . . . .	<a href="#">15</a>
	Acknowledgments. . . . .	<a href="#">16</a>
	Author's Addresses . . . . .	<a href="#">17</a>
	Full Copyright Statement . . . . .	<a href="#">18</a>

[1](#) Introduction

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](#)].

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

This document is a proposal put forward by the CRYPT-PRO Company to provide supplemental information and specifications needed by the "Russian Cryptographic Software Compatibility Agreement" community.

The algorithm identifiers and associated parameters 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, and the encoding format for the signatures produced by these algorithms are specified. 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

Internet 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 [\[CPALGS\]](#).

## [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 identifies OIDs and ASN.1 encoding for public keys contained in a certificate.

The conforming CAs and/or applications MUST fully support digital signatures and public keys for at least one of the specified algorithms.

### [2.1](#) One-way Hash Function

This section identifies the use of one-way, collision free hash function GOST R 34.11-94 – the only one that can be used in digital signature algorithms 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 the arbitrary finite bit length input. This document does not contain the full GOST R 34.11-94 specification, which can be found in [GOSTR3411] 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 parameters in the subjectPublicKeyInfo field of the certificate

of the issuer SHALL apply to the verification of the signature.

Signature algorithm GOST R 34.10-94 generates digital signature in the form of two 256-bit numbers  $r'$  and  $s$ . Its octet string representation consists of 64 octets, where first 32 octets contain big endian representation of  $s$  and second 32 octets contain big endian representation of  $r'$ .

Signature values in CMS [[CMS](#)] are represented as octet strings, and the output is used directly. However, signature values in certificates and CRLs [[PROFILE](#)] are represented as bit strings, and conversion is needed.

To convert a 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 parameters in the subjectPublicKeyInfo field of the certificate

of the issuer SHALL apply to the verification of the signature.

Signature algorithm GOST R 34.10-2001 generates digital signature in the form of two 256-bit numbers  $r'$  and  $s$ . Its octet string representation consists of 64 octets, where first 32 octets contain big endian representation of  $s$  and second 32 octets contain big endian representation of  $r'$ .

Signature values in CMS [[CMS](#)] are represented as octet strings, and the output is used directly. However, signature values in certificates and CRLs [[PROFILE](#)] are represented as bit strings, and conversion is needed.

To convert a 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.

## [2.3](#) 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 signature algorithm GOST R 34.10-94 [[GOSTR341094](#)] and for 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) }
```

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

When the id-GostR3410-94 algorithm identifier appears as the algorithm field in an AlgorithmIdentifier, the encoding MAY completely 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 (see section 8.1 of [\[CPALGS\]](#))

Absence of parameters SHALL be processed as described in [RFC 3280 \[PROFILE\]](#), section 6.1, that is, parameters are inherited from the issuer certificate if possible.

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  $Y = a^x \pmod{p}$ , where a and

p - parameters.

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

digitalSignature;  
nonRepudiation.  
keyEncipherment;  
keyAgreement.

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

encipherOnly;  
decipherOnly.

The keyUsage extension MUST NOT assert both encipherOnly and decipherOnly.

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

digitalSignature;  
nonRepudiation;  
keyCertSign;  
cRLSign.

### [2.3.2](#) GOST R 34.10-2001 Keys

GOST R 34.10-2001 public keys can be used for signature algorithm GOST R 34.10-2001 [[GOSTR341001](#)] and for 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)



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

When the id-GostR3410-2001 algorithm identifier appears as the algorithm field in an AlgorithmIdentifier, the encoding MAY completely 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 (see section 8.1 of [\[CPALGS\]](#))

Absence of parameters SHALL be processed as described in [RFC 3280 \[PROFILE\]](#), section 6.1, that is, parameters are inherited from the issuer certificate if possible.

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

GostR3410-2001-PublicKey MUST must contain 64 octets, where first 32 octets contain little endian representation of x and second 32 octets contain little endian representation of y. This corresponds to the binary representation of ( $\langle y \rangle_{256} || \langle x \rangle_{256}$ ) from [\[GOSTR341001\]](#), ch. 5.3.

If the keyUsage extension is present in an end-entity certificate, which contains a GOST R 34.10-2001 public key, the following values MAY be present:

- digitalSignature,
- nonRepudiation,
- keyEncipherment,
- keyAgreement.

If the keyAgreement or keyEncipherment extension is present in a certificate, the following values MAY be present:

- encipherOnly,
- decipherOnly.

The keyUsage extension MUST NOT assert both encipherOnly and decipherOnly.

If the keyUsage extension is present in an CA or CRL signer certificate which contains a GOST R 34.10-2001 public key, the following values MAY be present:

- digitalSignature,
- nonRepudiation,
- keyCertSign,
- cRLSign.

### [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 certificate is used as analogue to a manual signing, in the context of Russian Federal Digital Signature Law [[RFDSL](#)], certificate MUST contain keyUsage extension, it MUST be critical, and keyUsage MUST NOT include keyEncipherment and keyAgreement.

When certificate validity period (typically 5 years for end entities and 7 years for CAs in Russia) is not equal to the private key validity period (typically 15 months in Russia) it is RECOMMENDED to use private key usage period extension.

For security discussion concerning use of algorithm parameters, see section Security Considerations from [[CPALGS](#)].

[4.1](#) GOST R 34.10-94 Certificate

-----BEGIN CERTIFICATE-----

MIICCzCCABoCECM042BGlST0xwvklBguFuswCAYGKoUDAgIEMGkxHTAbBgNVBAMM  
FEdvc3RSMzQxMC05NCBleGFtcGxlMRIwEAYDVQQKDA1DcnlwdG9Qcm8xCzAJBgNV  
BAYTAlJVMScwJQYJKoZIhvcNAQkBFhhHb3N0UjM0MTAtOTRAZXhhbXBsZS5jb20w  
HhcnMDUwODE2MTIzMjUwWhcnMTUwODE2MTIzMjUwWjBpMR0wGwYDVQQDDBRhb3N0  
UjM0MTAtOTQgZXhhbXBsZTESMBAGA1UECgwJQ3J5cHRvUHJvMQswCQYDVQQGEwJS  
VTEnMCUGCSqGSIB3DQEJARYYR29zdFIZNDEwLTk0QGV4YW1wbGUuY29tMIGlMBwG  
BiqFAwICFDASBgqhQMCAiACBgqhQMCAh4BA4GEAASBgLuEZuF5nls02CyAfx0o  
GWZxV/6MVCUhr28wCyd3RpjG+0dVvrey85Ns0bVCNyaE4g0QiiQOHwxCTs7ESuo  
v2Y5MlyUi8Go/htjEvYJJYfMdRv05YmKCYJo01x3pg+2kBATjeM+fJyR1qwNCCw+  
eMG1wra3Ggqqi0WBkzIydvP7MAgGBiqFAwICBANBABHHCH4S3ALxAiMPr3aPRyqB  
g1DjB8zy5DEjiULic+HeIveF81W9l0xGkZxnrFjXBSqnjLeFKgF1hffX0AP7zUM=  
-----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 to  
0x22F785F355BD94EC46919C67AC58D7052AA78CB7852A017585F7D73803FBCD43  
and s equals to  
0x11C7087E12DC02F102232947768F472A818350E307CCF2E431238942C873E1DE

## 4.2 GOST R 34.10-2001 Certificate

-----BEGIN CERTIFICATE-----

```
MIIB0DCCAX8CECv1xh7CEb0Xx9zUYma0LiEwCAYGKoUDAgIDMG0xHzAdBgNVBAMM
Fkdvc3RSMzQxMC0yMDAxIGV4YW1wbGUxEjAQBgnVBAAoMCUNyeXB0b1BybzELMAkG
A1UEBhMCUxKTAnBgkqhkiG9w0BCQEWGkdvc3RSMzQxMC0yMDAxQGV4YW1wbGUu
Y29tMB4XDTA1MDgxNjE0MTgyMFoXDTE1MDgxNjE0MTgyMFowbTEfMB0GA1UEAwwW
R29zdFIZNDEwLTIwMDEgZXhhbXBsZTESMBAGA1UECgwJQ3J5cHRvUHJvMQswCQYD
VQQGEwJSVTEpMCcGCsGSIb3DQEJARYaR29zdFIZNDEwLTIwMDFAZXhhbXBsZS5j
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 to  
0x577E324FE70F2B6DF45C437A0305E5FD2C89318C13CD0875401A026075689584  
and y equals to  
0x601AEACABC660FDFB0CBC7567EBBA6EA8DE40FAE857C9AD0038895B916CCEB8F

In the signature of the above certificate, r' equals to  
0xC1DE176E8D1BEC71B593F3DD36935577688989176220F4DAB131D5B51C33DEE2

and s equals to  
0x3C2FC90944B727A9ECA7D5E9FB536DD2C3AA647C442EDEED3116454FBC543FDD



## 5 References

### 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);
- [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] "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", V. Popov, I. Kurepkin, S. Leontiev, September 2005, [draft-popov-cryptopro-cpalgs-04.txt](#) work in progress;
- [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.
- [PKALGS] L. Bassham, W. Polk, R. Housley, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", [RFC 3279](#), 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.

#### Informative references:

- [Schneier95] B. Schneier, Applied cryptography, second edition, John Wiley & Sons, Inc., 1995;
- [RFDSL] Russian Federal Digital Signature Law, 10 Jan 2002 N1-FZ
- [RFC2119] Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [CMS] Housley, R., "Cryptographic Message Syntax (CMS)", [RFC 3852](#), July 2004.

#### 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-Alpha. The goal 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.

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

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

Russ Hously (Vigil Security, LLC, housley@vigilsec.com) and Vasilij Sakharov (DEMOS Co Ltd, svp@dol.ru) for initiative creating this document.

Internet-Draft

Using GOST with PKIX

September 2005

Grigorij Chudov for navigating the IETF process for this document.

This document is based on a contribution of CRYPTO-PRO company. Any substantial use of the text from this document must reference CRYPTO-PRO. CRYPTO-PRO requests that all material mentioning or referencing this document identify this as "CRYPTO-PRO CPPK".

#### Author's Addresses

Serguei Leontiev  
CRYPTO-PRO  
38, Obraztsova,  
Moscow, 127018, Russian Federation  
EMail: lse@cryptopro.ru

Dennis Shefanovski  
DEMOS Co Ltd  
6/1, Ovchinnikovskaja naberezhnaya,  
Moscow, 113035, Russian Federation  
EMail: sdb@dol.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, Obrastsova,  
Moscow, 127018, Russian Federation  
EMail: izotov@nii.voskhod.ru

Elena Minaeva  
MD PREI  
build 3, 6A, Vtoroj Troitskij per.,

Leontiev & Shefanovski

Informational

[Page 17]

---

Internet-Draft

Using GOST with PKIX

September 2005

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

#### Disclaimer of Validity

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.

#### Full Copyright Statement

Copyright (C) The Internet Society (2005). 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.

#### Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.