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 ....................................................2
  1.1. Requirement Words ........................................3
2. Algorithm Support ...............................................3
  2.1. One-Way Hash Function ....................................3
    2.1.1. One-Way Hash Function GOST R 34.11-94 ...........3
  2.2. Signature Algorithms ....................................4
    2.2.1. Signature Algorithm GOST R 34.10-94 .............4
    2.2.2. Signature Algorithm GOST R 34.10-2001 ..........5
  2.3. Subject Public Key Algorithms ...........................5
    2.3.1. GOST R 34.10-94 Keys ................................6
    2.3.2. GOST R 34.10-2001 Keys ...........................8
3. Security Considerations ........................................9
4. Examples .....................................................10
  4.1. GOST R 34.10-94 Certificate ..........................10
  4.2. GOST R 34.10-2001 Certificate ........................12
5. Acknowledgements ............................................15
6. References ....................................................16
  6.1. Normative References ....................................16
  6.2. Informative References .................................17

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 [GOST341194] (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:

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

When the $\text{id-GostR3411-94-with-GostR3410-2001}$ 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 $\text{id-GostR3411-94-with-GostR3410-2001}$.

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:

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

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:

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

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 \( Y = a^x \pmod{p} \), 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 keyEncipherment 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-----
MIICCczCAbCCECMO42BgSTxwvk1BqfuswCAYGKoUDAgIEMGkxHTABgKVNBAMM
FE7dv3RSMzQxMC05NCBlGftcGxlM1RIwEAYDVQQKDA1DcnldG9Qcm8xCzAJBgNV
BAYTa1JVMScwQYJKoZIhvcNAQkBFhnhHb3NUUjM0MTAtOTRAZXhhbXBsZS5jb20w
HhcNMDUwODE2MTIzMjUwNMTUwODE2MTIzMjUwMjBpMR0wGwYDVQQDDBBRHb3N0
UjM0MTAtOTQxXhnhXBsZTESMBAGA1UwCgiegJQ3JscHRvUHJvMkQswCQYDVQQGEwJJS
VTEwMCUGCSqGSIb3DQEJARYYR29zdFJzNDEwLTk0QGV4Y1gwGUw29tMIQIcBoB
BiqFARIFCIASBgcghQMCAiACBgcghQMCAh4BA4GAEASBgLuEZuF5nl02CyAfxOo
GWZzYV/6MVCUhR28wCyd3RjyG0dVvrey85NaObVCNyE40gOlOHorxcTSS7ESuo
vY5M1yU1iGo/hjtjeVYJJYfMdrv05YkC1IoOs3p+2kBATjeM+fJyRlqWNNCCw+
eMGlwra3Gqqi0WBkzIydvp7MAgGBiqFAwICBANABBPw4S3ALxiAiMr3aPRyQB
91dj85yzDEj+ULiC+HeIveF8i1W910vGkZXnrFjXBSnqjLeFKgF1hffXOAP7zUM=
-----END CERTIFICATE-----
```

0 30 523: SEQUENCE {
4 30 442:  SEQUENCE { 
8 02 16:  INTEGER 
: 23 0E E3 40 66 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) 

Leontiev & Shefanovski Standards Track [Page 10]
IA5String 'GostR3410-94@example.com'

UTCTime '050816123250Z'

UTCTime '150816123250Z'

OBJECT IDENTIFIER commonName (2 5 4 3)

UTF8String 'GostR3410-94 example'

OBJECT IDENTIFIER organizationName (2 5 4 10)

UTF8String 'CryptoPro'

OBJECT IDENTIFIER countryName (2 5 4 6)

PrintableString 'RU'

OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)

IA5String 'GostR3410-94@example.com'

OBJECT IDENTIFIER id-GostR3410-94 (1 2 643 2 2 20)

OBJECT IDENTIFIER

id-GostR3410-94-CryptoPro-A-ParamSet

(1 2 643 2 2 32 2)

OBJECT IDENTIFIER

id-GostR3411-94-CryptoProParamSet

(1 2 643 2 2 30 1)

BIT STRING 0 unused bits, encapsulates {

OCTET STRING
In the signature of the above certificate, r’ equals
0x22F785F355BD94EC6919C578D02A78CB7852A017585F7D73803FBCD43
and s equals
0x11C7087E12DC02F102232947768F472A818350E307CCF2E431238942C873E1DE

4.2. GOST R 34.10-2001 Certificate

-----BEGIN CERTIFICATE-----
MIIB0DCCAX8CECv1xh7C7EboXXz9UyIoAwCAYGK0uDAgIDMGOxHzAdBgNVBAMMBkdv
3CMzM0ADAxIGV4YW1wGUXEjAQBgNVBAoMCUNyeXB0bGRvY25lZmFyb3V0MQswCQYD
VQQGEwJSVTEpMCcGCSqGSIb3DQEJARYaR29zdFIzNDEwLTIwMDFAZXhhbXBsZS5j
b20wYzAcBgYqhkOAwE0Y0BQMoMBAgdSfeRQ6C1nKjgYyG15X1d0M0Q2MDE2OTIu
WzQ1NjUyMTgwMjI1MDQyOTMxNjI5MjQ4OTA4MDAwMDEwNTg4MDE3MTI2MDEyMiwg
-----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

Leontiev & Shefanovski Standards Track
id-GostR3411-94-with-GostR3410-2001 (1 2 643 2 2 3)

SEQUENCE {
  SET {
    SEQUENCE {
      OBJECT IDENTIFIER commonName (2 5 4 3)
      UTF8String 'GostR3410-2001 example'
    }
  }
}

SET {
  SEQUENCE {
    OBJECT IDENTIFIER organizationName (2 5 4 10)
    UTF8String 'CryptoPro'
  }
}

SET {
  SEQUENCE {
    OBJECT IDENTIFIER countryName (2 5 4 6)
    PrintableString 'RU'
  }
}

SET {
  SEQUENCE {
    OBJECT IDENTIFIER emailAddress (1 2 840 113549 1 9 1)
    IA5String 'GostR3410-2001@example.com'
  }
}

SEQUENCE {
  UTCTime '050816141820Z'
  UTCTime '150816141820Z'
}

SEQUENCE {
  SEQUENCE {
    OBJECT IDENTIFIER commonName (2 5 4 3)
    UTF8String 'GostR3410-2001 example'
  }
}

SET {
  SEQUENCE {
    OBJECT IDENTIFIER organizationName (2 5 4 10)
    UTF8String 'CryptoPro'
  }
}

SET {
  SEQUENCE {
    OBJECT IDENTIFIER countryName (2 5 4 6)
  }
}
In the public key of the above certificate, x equals
0x577E324FE70F2B6DF45C437A0305E5FD2C89318C13CD0875401A026075689584
and y equals
0x601AEACABC660FDFB0CBB7567E8BA6E8DE40F857C9AD0038895B916CCEB8F
The corresponding private key d equals
0x0B293BE050D0082BDAE785631A6BAB68F35B42786D6DDA56AFAF169891040F77
In the signature of the above certificate, r equals
0xC1DE176E8D1BEC71B593F3DD36935577688989176220F4DAB131D5B51C33DEE2
and s equals
0x3C2FC9094B72A9ECA7D5E9FB536DD2C3AA647C442EDEED3116454FBC543FDD

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


[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)


6.2. Informative References


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

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, Leninskiy 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).