

Network Working Group
Request for Comments: 4357
Category: Informational

V. Popov
I. Kurepkin
S. Leontiev
CRYPTO-PRO
January 2006

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

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document describes the cryptographic algorithms and parameters supplementary to the original GOST specifications, GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94, for use in Internet applications.

Table of Contents

1. Introduction	2
1.1. Terminology	2
2. Cipher Modes and Parameters	3
2.1. GOST 28147-89 CBC Mode	4
2.2. GOST 28147-89 Padding Modes	4
2.3. Key Meshing Algorithms	4
2.3.1. Null Key Meshing	5
2.3.2. CryptoPro Key Meshing	5
3. HMAC_GOSTR3411	6
4. PRF_GOSTR3411	6
5. Key Derivation Algorithms	6
5.1. VKO GOST R 34.10-94	6
5.2. VKO GOST R 34.10-2001	7
6. Key Wrap Algorithms	7
6.1. GOST 28147-89 Key Wrap	7
6.2. GOST 28147-89 Key Unwrap	8
6.3. CryptoPro Key Wrap	8
6.4. CryptoPro Key Unwrap	9
6.5. CryptoPro KEK Diversification Algorithm	9

7. Secret Key Diversification	10
8. Algorithm Parameters	10
8.1. Encryption Algorithm Parameters	10
8.2. Digest Algorithm Parameters	11
8.3. GOST R 34.10-94 Public Key Algorithm Parameters	12
8.4. GOST R 34.10-2001 Public Key Algorithm Parameters	13
9. Security Considerations	14
10. Appendix ASN.1 Modules	15
10.1. Cryptographic-Gost-Useful-Definitions	15
10.2. Gost28147-89-EncryptionSyntax	17
10.3. Gost28147-89-ParamSetSyntax	19
10.4. GostR3411-94-DigestSyntax	21
10.5. GostR3411-94-ParamSetSyntax	22
10.6. GostR3410-94-PKISyntax	23
10.7. GostR3410-94-ParamSetSyntax	25
10.8. GostR3410-2001-PKISyntax	27
10.9. GostR3410-2001-ParamSetSyntax	29
11. Appendix Parameters	30
11.1. Encryption Algorithm Parameters	30
11.2. Digest Algorithm Parameters	33
11.3. GOST R 34.10-94 Public Key Algorithm Parameters	34
11.4. GOST R 34.10-2001 Public Key Algorithm Parameters	42
12. Acknowledgements	46
13. References	47
13.1. Normative References	47
13.2. Informative References	47

1. Introduction

Russian cryptographic standards that define the algorithms GOST 28147-89 [GOST28147], GOST R 34.10-94 [GOSTR341094], GOST R 34.10-2001 [GOSTR341001], and GOST R34.11-94 [GOSTR341194] provide basic information about how the algorithms work, but supplemental specifications are needed to effectively use the algorithms (a brief English technical description of these algorithms can be found in [Schneier95]).

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

1.1. Terminology

In this document, the key words MUST, MUST NOT, REQUIRED, SHOULD, SHOULD NOT, RECOMMENDED, and MAY are to be interpreted as described in [RFC2119].

The following functions and operators are also used in this document:

'|' stands for concatenation.

'~' stands for bitwise NOT operator.

'^' stands for the power operator.

encryptECB (K, D) is D, encrypted with key K using GOST 28147-89 in "prostaya zamena" (ECB) mode.

decryptECB (K, D) is D, decrypted with key K using GOST 28147-89 in ECB mode.

encryptCFB (IV, K, D) is D, encrypted with key K using GOST 28147-89 in "gammirovanie s obratnoj svyaziyu" (64-bit CFB) mode, and IV is used as the initialization vector.

encryptCNT (IV, K, D) is D, encrypted with key K using GOST 28147-89 in "gammirovanie" (counter) mode, and IV is used as the initialization vector.

gostR3411 (D) is the 256-bit result of the GOST R 34.11-94 hash function, used with zero initialization vector, and S-Box parameter, defined by id-GostR3411-94-CryptoProParamSet (see Section 11.2).

gost28147IMIT (IV, K, D) is the 32-bit result of the GOST 28147-89 in "imitovstavka" (MAC) mode, used with D as plaintext, K as key and IV as initialization vector. Note that the standard specifies its use in this mode only with an initialization vector of zero.

When keys and initialization vectors are converted to/from byte arrays, little-endian byte order is assumed.

2. Cipher Modes and Parameters

This document defines four cipher properties that allow an implementer to vary cipher operations. The four parameters are the cipher mode, the key meshing algorithm, the padding mode, and the S-box.

[GOST28147] defines only three cipher modes for GOST 28147-89: ECB, CFB, and counter mode. This document defines an additional cipher mode, CBC.

When GOST 28147-89 is used to process large amounts of data, a symmetric key should be protected by a key meshing algorithm. Key meshing transforms a symmetric key after some amount of data has been processed. This document defines the CryptoPro key meshing algorithm.

The cipher mode, key meshing algorithm, padding mode, and S-box are specified by algorithm parameters.

2.1. GOST 28147-89 CBC Mode

This section provides the supplemental information for GOST 28147-89 (a block-to-block primitive) needed to operate in CBC mode.

Before each plaintext block is encrypted, it is combined with the cipher text of the previous block via a bitwise XOR operation. This ensures that even if the plaintext contains many identical blocks, each block will encrypt to a different cipher text block. The initialization vector is combined with the first plaintext block by a bitwise XOR operation before the block is encrypted.

2.2. GOST 28147-89 Padding Modes

This section provides the supplemental information for GOST 28147-89, needed to operate on plaintext where the length is not divisible by GOST 28147-89 block size (8 bytes).

Let x ($0 < x \leq 8$) be the number of bytes in the last, possibly incomplete, block of data.

There are three padding modes:

- * Zero padding: $8-x$ remaining bytes are filled with zero
- * PKCS#5 padding: $8-x$ remaining bytes are filled with the value of $8-x$. If there's no incomplete block, one extra block filled with value 8 is added.
- * Random padding: $8-x$ remaining bytes of the last block are set to random.

2.3. Key Meshing Algorithms

Key meshing algorithms transform the key after processing a certain amount of data. In applications that must be strictly robust to attacks based on timing and EMI analysis, one symmetric key should not be used for quantities of plaintext larger than 1024 octets.

A key meshing algorithm affects internal cipher state; it is not a protocol level feature. Its role is similar to that of a cipher mode. The choice of key meshing algorithm is usually dictated by the encryption algorithm parameters, but some protocols explicitly specify applicable key meshing algorithms.

All encryption parameter sets defined in this document specify the use of the CryptoPro key meshing algorithm, except for id-Gost28147-89-TestParamSet, which specifies use of null key meshing algorithm.

2.3.1. Null Key Meshing

The null key meshing algorithm never changes a key.

The identifier for this algorithm is:

```
id-Gost28147-89-None-KeyMeshing OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
    keyMeshing(14) none(0) }
```

There are no meaningful parameters to this algorithm. If present, AlgorithmIdentifier.parameters MUST contain NULL.

2.3.2. CryptoPro Key Meshing

The CryptoPro key meshing algorithm transforms the key and initialization vector every 1024 octets (8192 bits, or 256 64-bit blocks) of plaintext data.

This algorithm has the same drawback as OFB cipher mode: it is impossible to re-establish crypto synch while decrypting a ciphertext if parts of encrypted data are corrupted, lost, or processed out of order. Furthermore, it is impossible to re-synch even if an IV for each data packet is provided explicitly. Use of this algorithm in protocols such as IPsec ESP requires special care.

The identifier for this algorithm is:

```
id-Gost28147-89-CryptoPro-KeyMeshing OBJECT IDENTIFIER ::=
  { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
    keyMeshing(14) cryptoPro(1) }
```

There are no meaningful parameters to this algorithm. If present, AlgorithmIdentifier.parameters MUST contain NULL.

GOST 28147-89, in encrypt, decrypt, or MAC mode, starts with key $K[0] = K$, $IV_0[0] = IV$, $i = 0$. Let $IV_n[0]$ be the value of the initialization vector after processing the first 1024 octets of data.

Processing of the next 1024 octets will start with $K[1]$ and $IV0[1]$, which are calculated using the following formula:

```
K[i+1] = decryptECB (K[i], C);
IV0[i+1] = encryptECB (K[i+1], IVn[i])
```

Where $C = \{0x69, 0x00, 0x72, 0x22, 0x64, 0xC9, 0x04, 0x23,$
 $0x8D, 0x3A, 0xDB, 0x96, 0x46, 0xE9, 0x2A, 0xC4,$
 $0x18, 0xFE, 0xAC, 0x94, 0x00, 0xED, 0x07, 0x12,$
 $0xC0, 0x86, 0xDC, 0xC2, 0xEF, 0x4C, 0xA9, 0x2B\};$

After processing each 1024 octets of data:

- * the resulting initialization vector is stored as $IVn[i]$;
- * $K[i+1]$ and $IV0[i+1]$ are calculated;
- * i is incremented;
- * Encryption or decryption of next 1024 bytes starts, using the new key and IV;

The process is repeated until all the data has been processed.

3. HMAC_GOSTR3411

HMAC_GOSTR3411 (K, text) function is based on the hash function GOST R 34.11-94, as defined in [HMAC], with the following parameter values:
 $B = 32, L = 32.$

4. PRF_GOSTR3411

PRF_GOSTR3411 is a pseudorandom function, based on HMAC_GOSTR3411. It is calculated as P_{hash} , defined in Section 5 of [TLS].
 $\text{PRF_GOSTR3411}(\text{secret}, \text{label}, \text{seed}) = P_{\text{GOSTR3411}}(\text{secret}, \text{label} | \text{seed}).$

5. Key Derivation Algorithms

Standards [GOSTR341094] and [GOSTR341001] do not define any key derivation algorithms.

Section 5.1 specifies algorithm VKO GOST R 34.10-94, which generates GOST KEK using two GOST R 34.10-94 keypairs.

Section 5.2 specifies algorithm VKO GOST R 34.10-2001, which generates GOST KEK using two GOST R 34.10-2001 keypairs and UKM.

Keypairs MUST have identical parameters.

5.1. VKO GOST R 34.10-94

This algorithm creates a key encryption key (KEK) using the sender's private key and the recipient's public key (or vice versa).

Exchange key KEK is a 256-bit hash of the 1024-bit shared secret that is generated using Diffie-Hellman key agreement.

- 1) Let $K(x,y) = a^{(x*y)} \pmod{p}$, where
 - x - sender's private key, a^x - sender's public key
 - y - recipient's private key, a^y - recipient's public key
 - a, p - parameters
- 2) Calculate a 256-bit hash of $K(x,y)$:
 - $KEK(x,y) = \text{gostR3411}(K(x,y))$

Keypairs (x, a^x) and (y, a^y) MUST comply with [GOSTR341094].

This algorithm MUST NOT be used when $a^x = a \pmod{p}$ or $a^y = a \pmod{p}$.

5.2. VKO GOST R 34.10-2001

This algorithm creates a key encryption key (KEK) using 64 bit UKM, the sender's private key, and the recipient's public key (or the reverse of the latter pair).

- 1) Let $K(x,y,UKM) = ((UKM*x) \pmod{q}) \cdot (y.P)$ (512 bit), where
 - x - sender's private key (256 bit)
 - x.P - sender's public key (512 bit)
 - y - recipient's private key (256 bit)
 - y.P - recipient's public key (512 bit)
 - UKM - non-zero integer, produced as in step 2 p. 6.1 [GOSTR341001]
 - P - base point on the elliptic curve (two 256-bit coordinates)
 - $UKM*x$ - x multiplied by UKM as integers
 - x.P - a multiple point
- 2) Calculate a 256-bit hash of $K(x,y,UKM)$:
 - $KEK(x,y,UKM) = \text{gostR3411}(K(x,y,UKM))$

Keypairs $(x, x.P)$ and $(y, y.P)$ MUST comply with [GOSTR341001].

This algorithm MUST NOT be used when $x.P = P$, $y.P = P$

6. Key Wrap Algorithms

This document defines two key wrap algorithms: GOST 28147-89 Key Wrap and CryptoPro Key Wrap. These are used to encrypt a Content Encryption Key (CEK) with a Key Encryption Key (KEK).

6.1. GOST 28147-89 Key Wrap

This algorithm encrypts GOST 28147-89 CEK with a GOST 28147-89 KEK.

Note: This algorithm MUST NOT be used with a KEK produced by VKO GOST R 34.10-94, because such a KEK is constant for every sender-recipient pair. Encrypting many different content encryption keys on the same constant KEK may reveal that KEK.

The GOST 28147-89 key wrap algorithm is:

- 1) For a unique symmetric KEK, generate 8 octets at random and call the result UKM. For a KEK, produced by VKO GOST R 34.10-2001, use the UKM that was used for key derivation.
- 2) Compute a 4-byte checksum value, gost28147IMIT (UKM, KEK, CEK). Call the result CEK_MAC.
- 3) Encrypt the CEK in ECB mode using the KEK. Call the ciphertext CEK_ENC.
- 4) The wrapped content-encryption key is (UKM | CEK_ENC | CEK_MAC).

6.2. GOST 28147-89 Key Unwrap

This algorithm decrypts GOST 28147-89 CEK with a GOST 28147-89 KEK. The GOST 28147-89 key unwrap algorithm is:

- 1) If the wrapped content-encryption key is not 44 octets, then error.
- 2) Decompose the wrapped content-encryption key into UKM, CEK_ENC, and CEK_MAC. UKM is the most significant (first) 8 octets. CEK_ENC is next 32 octets, and CEK_MAC is the least significant (last) 4 octets.
- 3) Decrypt CEK_ENC in ECB mode using the KEK. Call the output CEK.
- 4) Compute a 4-byte checksum value, gost28147IMIT (UKM, KEK, CEK), compare the result with CEK_MAC. If they are not equal, then error.

6.3. CryptoPro Key Wrap

This algorithm encrypts GOST 28147-89 CEK with a GOST 28147-89 KEK. It can be used with any KEK (e.g., produced by VKO GOST R 34.10-94 or VKO GOST R 34.10-2001) because a unique UKM is used to diversify the KEK.

The CryptoPro key wrap algorithm is:

- 1) For a unique symmetric KEK or a KEK produced by VKO GOST R 34.10-94, generate 8 octets at random. Call the result UKM. For a KEK, produced by VKO GOST R 34.10-2001, use the UKM that was used for key derivation.
- 2) Diversify KEK, using the CryptoPro KEK Diversification Algorithm, described in Section 6.5. Call the result KEK(UKM).

- 3) Compute a 4-byte checksum value, gost28147IMIT (UKM, KEK(UKM), CEK). Call the result CEK_MAC.
- 4) Encrypt CEK in ECB mode using KEK(UKM). Call the ciphertext CEK_ENC.
- 5) The wrapped content-encryption key is (UKM | CEK_ENC | CEK_MAC).

6.4. CryptoPro Key Unwrap

This algorithm encrypts GOST 28147-89 CEK with a GOST 28147-89 KEK. The CryptoPro key unwrap algorithm is:

- 1) If the wrapped content-encryption key is not 44 octets, then it is an error.
- 2) Decompose the wrapped content-encryption key into UKM, CEK_ENC, and CEK_MAC. UKM is the most significant (first) 8 octets. CEK_ENC is next 32 octets, and CEK_MAC is the least significant (last) 4 octets.
- 3) Diversify KEK using the CryptoPro KEK Diversification Algorithm, described in section 6.5. Call the result KEK(UKM).
- 4) Decrypt CEK_ENC in ECB mode using KEK(UKM). Call the output CEK.
- 5) Compute a 4-byte checksum value, gost28147IMIT (UKM, KEK(UKM), CEK), compare the result with CEK_MAC. If they are not equal, then it is an error.

6.5. CryptoPro KEK Diversification Algorithm

Given a random 64-bit UKM and a GOST 28147-89 key K, this algorithm creates a new GOST 28147-89 key K(UKM).

- 1) Let $K[0] = K$;
- 2) UKM is split into components $a[i,j]$:
 $UKM = a[0] || \dots || a[7]$ ($a[i]$ - byte, $a[i,0] \dots a[i,7]$ - it's bits)
- 3) Let i be 0.
- 4) $K[1] \dots K[8]$ are calculated by repeating the following algorithm eight times:
 - A) $K[i]$ is split into components $k[i,j]$:
 $K[i] = k[i,0] || k[i,1] || \dots || k[i,7]$ ($k[i,j]$ - 32-bit integer)
 - B) Vector $S[i]$ is calculated:
 $S[i] = ((a[i,0]*k[i,0] + \dots + a[i,7]*k[i,7]) \bmod 2^{32}) |$
 $((\sim a[i,0])*k[i,0] + \dots + (\sim a[i,7])*k[i,7]) \bmod 2^{32});$
 - C) $K[i+1] = \text{encryptCFB}(S[i], K[i], K[i])$
 - D) $i = i + 1$
- 5) Let $K(UKM)$ be $K[8]$.

7. Secret Key Diversification

This algorithm creates a GOST 28147-89 key K_d , given GOST R 34.10-94 or GOST R 34.10-2001 secret key K and diversification data D of size 4..40 bytes.

- 1) 40-byte blob B is created from D by cloning it enough times to fill all 40 bytes. For example, if D is 40-bytes long, $B = D$; If D is 6-bytes long, $B = D|D|D|D|D|D|D[0..3]$.
- 2) B is split into 8-byte UKM and 32-byte $SRCKEY$ ($B = UKM|SRCKEY$).
- 3) The algorithm from Section 6.5 is used to create $K(UKM)$ from key K and UKM , with two differences:
 - * Instead of $S[i]$, vector $(0,0,0,UKM[i],ff,ff,ff,ff \text{ XOR } UKM[i])$ is used.
 - * During each encryption step, only 8 out of 32 GOST 28147-89 rounds are done.
- 4) K_d is calculated:
 $K_d = \text{encryptCFB}(UKM, K(UKM), SRCKEY)$.

8. Algorithm Parameters

Standards [GOST28147], [GOST341194], [GOSTR341094], and [GOSTR341001] do not define specific values for algorithm parameters.

This document introduces the use of ASN.1 object identifiers (OIDs) to specify algorithm parameters.

Identifiers for all of the proposed parameter sets can be found in Appendix ASN.1 modules. Corresponding parameter values for proposed parameter sets can be found in Section 11.

8.1. Encryption Algorithm Parameters

GOST 28147-89 can be used in several modes; additional CBC mode is defined in Section 2.1. It also has an S-Box parameter. (See the Algorithm Parameters part in [GOST28147] in Russian; for a description in English, see [Schneier95], ch. 14.1, p. 331.)

This table contains the list of proposed parameter sets for GOST 28147-89:

```
Gost28147-89-ParamSetAlgorithms ALGORITHM-IDENTIFIER ::= {
  { Gost28147-89-ParamSetParameters IDENTIFIED BY
    id-Gost28147-89-TestParamSet } |
  { Gost28147-89-ParamSetParameters IDENTIFIED BY
    id-Gost28147-89-CryptoPro-A-ParamSet } |
  { Gost28147-89-ParamSetParameters IDENTIFIED BY
    id-Gost28147-89-CryptoPro-B-ParamSet } |
```

```

    { Gost28147-89-ParamSetParameters IDENTIFIED BY
      id-Gost28147-89-CryptoPro-C-ParamSet } |
    { Gost28147-89-ParamSetParameters IDENTIFIED BY
      id-Gost28147-89-CryptoPro-D-ParamSet }
  }

```

Identifier values are in the Appendix ASN.1 modules, and corresponding parameters are in Section 11.1.

Parameters for GOST 28147-89 are presented in the following form:

```

Gost28147-89-ParamSetParameters ::= SEQUENCE {
  eUZ          Gost28147-89-UZ,
  mode         INTEGER {
    gost28147-89-CNT(0),
    gost28147-89-CFB(1),
    cryptoPro-CBC(2)
  },
  shiftBits   INTEGER { gost28147-89-block(64) },
  keyMeshing  AlgorithmIdentifier
}
Gost28147-89-UZ ::= OCTET STRING (SIZE (64))
Gost28147-89-KeyMeshingAlgorithms ALGORITHM-IDENTIFIER ::= {
  { NULL IDENTIFIED BY id-Gost28147-89-CryptoPro-KeyMeshing } |
  { NULL IDENTIFIED BY id-Gost28147-89-None-KeyMeshing }
}

```

where

```

eUZ          - S-box value;
mode         - cipher mode;
shiftBits   - cipher parameter;
keyMeshing  - key meshing algorithm identifier.

```

8.2. Digest Algorithm Parameters

This table contains the list of proposed parameter sets for [GOST341194]:

```

GostR3411-94-ParamSetAlgorithms ALGORITHM-IDENTIFIER ::= {
  { GostR3411-94-ParamSetParameters IDENTIFIED BY
    id-GostR3411-94-TestParamSet
  } |
  { GostR3411-94-ParamSetParameters IDENTIFIED BY
    id-GostR3411-94-CryptoProParamSet
  }
}

```

Identifier values are in the Appendix ASN.1 modules, and corresponding parameters are in Section 11.2.

Parameters for [GOST341194] are presented in the following form:

```
GostR3411-94-ParamSetParameters ::=
  SEQUENCE {
    hUZ Gost28147-89-UZ,    -- S-Box for digest
    h0  GostR3411-94-Digest -- start digest value
  }
GostR3411-94-Digest ::= OCTET STRING (SIZE (32))
```

8.3. GOST R 34.10-94 Public Key Algorithm Parameters

This table contains the list of proposed parameter sets for GOST R 34.10-94:

```
GostR3410-94-ParamSetAlgorithm ALGORITHM-IDENTIFIER ::= {
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-TestParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-A-ParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-B-ParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-C-ParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-D-ParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-XchA-ParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-XchB-ParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-XchC-ParamSet }
}
```

Identifier values are in the Appendix ASN.1 modules, and corresponding parameters are in Section 11.3.

Parameters for GOST R 34.10-94 are presented in the following form:

```
GostR3410-94-ParamSetParameters ::=
  SEQUENCE {
    t      INTEGER,
    p      INTEGER,
    q      INTEGER,
    a      INTEGER,
    validationAlgorithm  AlgorithmIdentifier {{
```

```

    GostR3410-94-ValidationAlgorithms
    }} OPTIONAL
}

```

```

GostR3410-94-ValidationParameters ::=
    SEQUENCE {
        x0      INTEGER,
        c       INTEGER,
        d       INTEGER OPTIONAL
    }

```

Where

t - bit length of p (512 or 1024 bits);
p - modulus, prime number, $2^{(t-1)} < p < 2^t$;
q - order of cyclic group, prime number, $2^{254} < q < 2^{256}$, q is a factor of p-1;
a - generator, integer, $1 < a < p-1$, at that $aq \pmod p = 1$;
validationAlgorithm - constant p, q and a calculating algorithm.

x0 - seed;
c - used for p and q generation;
d - used for a generation.

8.4. GOST R 34.10-2001 Public Key Algorithm Parameters

This table contains the list of proposed parameter sets for GOST R 34.10-2001:

```

GostR3410-2001-ParamSetAlgorithm ALGORITHM-IDENTIFIER ::= {
    { GostR3410-2001-ParamSetParameters IDENTIFIED BY
      id-GostR3410-2001-TestParamSet } |
    { GostR3410-2001-ParamSetParameters IDENTIFIED BY
      id-GostR3410-2001-CryptoPro-A-ParamSet } |
    { GostR3410-2001-ParamSetParameters IDENTIFIED BY
      id-GostR3410-2001-CryptoPro-B-ParamSet } |
    { GostR3410-2001-ParamSetParameters IDENTIFIED BY
      id-GostR3410-2001-CryptoPro-C-ParamSet } |
    { GostR3410-2001-ParamSetParameters IDENTIFIED BY
      id-GostR3410-2001-CryptoPro-XchA-ParamSet } |
    { GostR3410-2001-ParamSetParameters IDENTIFIED BY
      id-GostR3410-2001-CryptoPro-XchB-ParamSet }
}

```

Identifier values are in the Appendix ASN.1 modules, and corresponding parameters are in Section 11.4.

Parameters for GOST R 34.10-2001 are presented in the following form:

```
GostR3410-2001-ParamSetParameters ::=
  SEQUENCE {
    a      INTEGER,
    b      INTEGER,
    p      INTEGER,
    q      INTEGER,
    x      INTEGER,
    y      INTEGER
  }
```

a, b - coefficients a and b of the elliptic curve E;
p - prime number - elliptic curve modulus;
q - prime number - order of cyclic group;
x, y - base point p coordinates.

9. Security Considerations

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

Cryptographic algorithm parameters affect rigidity of algorithms. The algorithm parameters proposed and described herein, except for the test parameter sets (id-Gost28147-89-TestParamSet, id-GostR3411-94-TestParamSet, id-GostR3410-94-TestParamSet, id-GostR3410-2001-TestParamSet), have been analyzed by a special certification laboratory of Scientific and Technical Center, "ATLAS", and by the Center of Certification Investigations in appropriate levels of target_of_evaluation (TOE), according to [RFDSL], [RFLIC], and [CRYPTOLIC].

Use of the test parameter sets or parameter sets not described herein is NOT RECOMMENDED. When different parameters are used, it is RECOMMENDED that they be subjected to examination by an authorized agency with approved methods of cryptographic analysis.

10. Appendix ASN.1 Modules

10.1. Cryptographic-Gost-Useful-Definitions

```
Cryptographic-Gost-Useful-Definitions
```

```
{ iso(1) member-body(2) ru(643) rans(2)
  cryptopro(2) other(1) modules(1)
  cryptographic-Gost-Useful-Definitions(0) 1 }
```

```
DEFINITIONS ::=
```

```
BEGIN
```

```
-- EXPORTS All --
```

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

```
-- Crypto-Pro OID branch
```

```
id-CryptoPro OBJECT IDENTIFIER ::=
```

```
{ iso(1) member-body(2) ru(643) rans(2) cryptopro(2) }
```

```
id-CryptoPro-algorithms OBJECT IDENTIFIER ::=
```

```
id-CryptoPro
```

```
id-CryptoPro-modules OBJECT IDENTIFIER ::=
```

```
{ id-CryptoPro other(1) modules(1) }
```

```
id-CryptoPro-hashes OBJECT IDENTIFIER ::=
```

```
{ id-CryptoPro-algorithms hashes(30) }
```

```
id-CryptoPro-encrypts OBJECT IDENTIFIER ::=
```

```
{ id-CryptoPro-algorithms encrypts(31) }
```

```
id-CryptoPro-signs OBJECT IDENTIFIER ::=
```

```
{ id-CryptoPro-algorithms signs(32) }
```

```
id-CryptoPro-exchanges OBJECT IDENTIFIER ::=
```

```
{ id-CryptoPro-algorithms exchanges(33) }
```

```
id-CryptoPro-extensions OBJECT IDENTIFIER ::=
```

```
{ id-CryptoPro extensions(34) }
```

```
id-CryptoPro-ecc-signs OBJECT IDENTIFIER ::=
```

```
{ id-CryptoPro-algorithms ecc-signs(35) }
```

```
id-CryptoPro-ecc-exchanges OBJECT IDENTIFIER ::=
```

```
{ id-CryptoPro-algorithms ecc-exchanges(36) }
```

```
id-CryptoPro-private-keys OBJECT IDENTIFIER ::=
```

```
{ id-CryptoPro-algorithms private-keys(37) }
```

```
id-CryptoPro-policyIds OBJECT IDENTIFIER ::=
```

```
{ id-CryptoPro policyIds(38) }
```

```
id-CryptoPro-policyQt OBJECT IDENTIFIER ::=
```

```
{ id-CryptoPro policyQt(39) }
```

```
id-CryptoPro-pkixcmp-infos OBJECT IDENTIFIER ::=
```

```

    { id-CryptoPro-algorithms pkixcmp-infos(41) }
id-CryptoPro-audit-service-types OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms audit-service-types(42) }
id-CryptoPro-audit-record-types OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms audit-record-types(43) }
id-CryptoPro-attributes OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms attributes(44) }
id-CryptoPro-name-service-types OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms name-service-types(45) }

-- ASN.1 modules of Russian Cryptography "GOST" & "GOST R"
-- Specifications
cryptographic-Gost-Useful-Definitions OBJECT IDENTIFIER ::=
    { id-CryptoPro-modules
      cryptographic-Gost-Useful-Definitions(0) 1 }
-- GOST R 34.11-94

gostR3411-94-DigestSyntax OBJECT IDENTIFIER ::=
    { id-CryptoPro-modules gostR3411-94-DigestSyntax(1) 1 }
gostR3411-94-ParamSetSyntax OBJECT IDENTIFIER ::=
    { id-CryptoPro-modules gostR3411-94-ParamSetSyntax(7) 1 }
-- GOST R 34.10-94

gostR3410-94-PKISyntax OBJECT IDENTIFIER ::=
    { id-CryptoPro-modules gostR3410-94-PKISyntax(2) 1 }
gostR3410-94-SignatureSyntax OBJECT IDENTIFIER ::=
    { id-CryptoPro-modules gostR3410-94-SignatureSyntax(3) 1 }
gostR3410-EncryptionSyntax OBJECT IDENTIFIER ::=
    { id-CryptoPro-modules gostR3410-EncryptionSyntax(5) 2 }
gostR3410-94-ParamSetSyntax OBJECT IDENTIFIER ::=
    { id-CryptoPro-modules gostR3410-94-ParamSetSyntax(8) 1 }
-- GOST R 34.10-2001

gostR3410-2001-PKISyntax OBJECT IDENTIFIER ::=
    { id-CryptoPro-modules gostR3410-2001-PKISyntax(9) 1 }
gostR3410-2001-SignatureSyntax OBJECT IDENTIFIER ::=
    { id-CryptoPro-modules
      gostR3410-2001-SignatureSyntax(10) 1 }
gostR3410-2001-ParamSetSyntax OBJECT IDENTIFIER ::=
    { id-CryptoPro-modules
      gostR3410-2001-ParamSetSyntax(12) 1 }
-- GOST 28147-89

gost28147-89-EncryptionSyntax OBJECT IDENTIFIER ::=
    { id-CryptoPro-modules gost28147-89-EncryptionSyntax(4) 1 }
gost28147-89-ParamSetSyntax OBJECT IDENTIFIER ::=
    { id-CryptoPro-modules gost28147-89-ParamSetSyntax(6) 1 }
-- Extended Key Usage for Crypto-Pro

```



```

gost-CryptoPro-ExtendedKeyUsage OBJECT IDENTIFIER ::=
  { id-CryptoPro-modules
    gost-CryptoPro-ExtendedKeyUsage(13) 1 }
-- Crypto-Pro Private keys

gost-CryptoPro-PrivateKey OBJECT IDENTIFIER ::=
  { id-CryptoPro-modules gost-CryptoPro-PrivateKey(14) 1 }
-- Crypto-Pro PKIXCMP structures

gost-CryptoPro-PKIXCMP OBJECT IDENTIFIER ::=
  { id-CryptoPro-modules gost-CryptoPro-PKIXCMP(15) 1 }
-- Crypto-Pro Transport Layer Security structures
gost-CryptoPro-TLS OBJECT IDENTIFIER ::=
  { id-CryptoPro-modules gost-CryptoPro-TLS(16) 1 }

-- Crypto-Pro Policy
gost-CryptoPro-Policy OBJECT IDENTIFIER ::=
  { id-CryptoPro-modules gost-CryptoPro-Policy(17) 1 }
gost-CryptoPro-Constants OBJECT IDENTIFIER ::=
  { id-CryptoPro-modules gost-CryptoPro-Constants(18) 1 }
-- Useful types
ALGORITHM-IDENTIFIER ::= CLASS {
  &id    OBJECT IDENTIFIER UNIQUE,
  &Type  OPTIONAL
}
WITH SYNTAX { [&Type] IDENTIFIED BY &id }
END -- Cryptographic-Gost-Useful-Definitions

```

10.2. Gost28147-89-EncryptionSyntax

```

Gost28147-89-EncryptionSyntax
  { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
    other(1) modules(1) gost28147-89-EncryptionSyntax(4) 1 }
DEFINITIONS EXPLICIT TAGS ::=
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for
-- use in the other ASN.1 modules contained within the Russian
-- Cryptography "GOST" & "GOST R" Specifications, and for the use
-- of other applications that will use them to access Russian
-- Cryptography services. Other applications may use them for
-- their own purposes, but this will not constrain extensions and
-- modifications needed to maintain or improve the Russian
-- Cryptography service.
  IMPORTS
    id-CryptoPro-algorithms, id-CryptoPro-encrypts,
    ALGORITHM-IDENTIFIER,
    cryptographic-Gost-Useful-Definitions

```

```

FROM Cryptographic-Gost-Useful-Definitions
  { iso(1) member-body(2) ru(643) rans(2)
    cryptopro(2) other(1) modules(1)
    cryptographic-Gost-Useful-Definitions(0) 1 }
;
-- GOST 28147-89 OID
id-Gost28147-89 OBJECT IDENTIFIER ::=
  { id-CryptoPro-algorithms gost28147-89(21) }
id-Gost28147-89-MAC OBJECT IDENTIFIER ::=
  { id-CryptoPro-algorithms gost28147-89-MAC(22) }
-- GOST 28147-89 cryptographic parameter sets OIDs
id-Gost28147-89-TestParamSet OBJECT IDENTIFIER ::=
  { id-CryptoPro-encrypts test(0) }
id-Gost28147-89-CryptoPro-A-ParamSet OBJECT IDENTIFIER ::=
  { id-CryptoPro-encrypts cryptopro-A(1) }
id-Gost28147-89-CryptoPro-B-ParamSet OBJECT IDENTIFIER ::=
  { id-CryptoPro-encrypts cryptopro-B(2) }
id-Gost28147-89-CryptoPro-C-ParamSet OBJECT IDENTIFIER ::=
  { id-CryptoPro-encrypts cryptopro-C(3) }
id-Gost28147-89-CryptoPro-D-ParamSet OBJECT IDENTIFIER ::=
  { id-CryptoPro-encrypts cryptopro-D(4) }
id-Gost28147-89-CryptoPro-Oscar-1-1-ParamSet
  OBJECT IDENTIFIER ::=
  { id-CryptoPro-encrypts cryptopro-Oscar-1-1(5) }
id-Gost28147-89-CryptoPro-Oscar-1-0-ParamSet
  OBJECT IDENTIFIER ::=
  { id-CryptoPro-encrypts cryptopro-Oscar-1-0(6) }
id-Gost28147-89-CryptoPro-RIC-1-ParamSet
  OBJECT IDENTIFIER ::=
  { id-CryptoPro-encrypts cryptopro-RIC-1(7) }
-- GOST 28147-89 Types
Gost28147-89-UZ ::= OCTET STRING (SIZE (64))
Gost28147-89-IV ::= OCTET STRING (SIZE (8))
Gost28147-89-Key ::= OCTET STRING (SIZE (32))
Gost28147-89-MAC ::= OCTET STRING (SIZE (1..4))
Gost28147-89-EncryptedKey ::=
  SEQUENCE {
    encryptedKey Gost28147-89-Key,
    maskKey      [0] IMPLICIT Gost28147-89-Key OPTIONAL,
    macKey       Gost28147-89-MAC (SIZE (4))
  }
Gost28147-89-ParamSet ::=
  OBJECT IDENTIFIER (
    id-Gost28147-89-TestParamSet |
      -- Only for testing purposes
    id-Gost28147-89-CryptoPro-A-ParamSet |
    id-Gost28147-89-CryptoPro-B-ParamSet |
    id-Gost28147-89-CryptoPro-C-ParamSet |
  )

```

```

        id-Gost28147-89-CryptoPro-D-ParamSet |
        id-Gost28147-89-CryptoPro-Oscar-1-1-ParamSet |
        id-Gost28147-89-CryptoPro-Oscar-1-0-ParamSet |
        id-Gost28147-89-CryptoPro-RIC-1-ParamSet
    )
    Gost28147-89-BlobParameters ::=
        SEQUENCE {
            encryptionParamSet Gost28147-89-ParamSet,
            ...
        }
    -- GOST 28147-89 encryption algorithm parameters
    Gost28147-89-Parameters ::=
        SEQUENCE {
            iv Gost28147-89-IV,
            encryptionParamSet Gost28147-89-ParamSet
        }
    Gost28147-89-Algorithms ALGORITHM-IDENTIFIER ::= {
        { Gost28147-89-Parameters IDENTIFIED BY
            id-Gost28147-89 }
    }
END -- Gost28147-89-EncryptionSyntax

```

10.3. Gost28147-89-ParamSetSyntax

```

Gost28147-89-ParamSetSyntax
    { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
      other(1) modules(1) gost28147-89-ParamSetSyntax(6) 1 }
DEFINITIONS EXPLICIT TAGS ::=
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for
-- use in the other ASN.1 modules contained within the Russian
-- Cryptography "GOST" & "GOST R" Specifications, and for the use
-- of other applications that will use them to access Russian
-- Cryptography services. Other applications may use them for
-- their own purposes, but this will not constrain extensions and
-- modifications needed to maintain or improve the Russian
-- Cryptography service.
IMPORTS
    id-CryptoPro-algorithms, id-CryptoPro-encrypts,
    gost28147-89-EncryptionSyntax, ALGORITHM-IDENTIFIER,
    cryptographic-Gost-Useful-Definitions
FROM Cryptographic-Gost-Useful-Definitions
    { iso(1) member-body(2) ru(643) rans(2)
      cryptopro(2) other(1) modules(1)
      cryptographic-Gost-Useful-Definitions(0) 1 }
    Gost28147-89-UZ,
    Gost28147-89-ParamSet,

```

```

id-Gost28147-89-TestParamSet,
id-Gost28147-89-CryptoPro-A-ParamSet,
id-Gost28147-89-CryptoPro-B-ParamSet,
id-Gost28147-89-CryptoPro-C-ParamSet,
id-Gost28147-89-CryptoPro-D-ParamSet
FROM Gost28147-89-EncryptionSyntax
    gost28147-89-EncryptionSyntax
AlgorithmIdentifier
FROM PKIX1Explicit88 {iso(1) identified-organization(3)
dod(6) internet(1) security(5) mechanisms(5) pkix(7)
id-mod(0) id-pkix1-explicit-88(1)}
;
-- GOST 28147-89 cryptographic parameter sets:
-- OIDs for parameter sets are imported from
-- Gost28147-89-EncryptionSyntax
Gost28147-89-ParamSetParameters ::=
SEQUENCE {
    eUZ          Gost28147-89-UZ,
    mode         INTEGER {
        gost28147-89-CNT(0),
        gost28147-89-CFB(1),
        cryptoPro-CBC(2)
    },
    shiftBits    INTEGER { gost28147-89-block(64) },
    keyMeshing   AlgorithmIdentifier
}
Gost28147-89-ParamSetAlgorithms ALGORITHM-IDENTIFIER ::= {
    { Gost28147-89-ParamSetParameters IDENTIFIED BY
      id-Gost28147-89-TestParamSet } |
    { Gost28147-89-ParamSetParameters IDENTIFIED BY
      id-Gost28147-89-CryptoPro-A-ParamSet } |
    { Gost28147-89-ParamSetParameters IDENTIFIED BY
      id-Gost28147-89-CryptoPro-B-ParamSet } |
    { Gost28147-89-ParamSetParameters IDENTIFIED BY
      id-Gost28147-89-CryptoPro-C-ParamSet } |
    { Gost28147-89-ParamSetParameters IDENTIFIED BY
      id-Gost28147-89-CryptoPro-D-ParamSet }
}
id-Gost28147-89-CryptoPro-KeyMeshing OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms keyMeshing(14) cryptoPro(1) }
id-Gost28147-89-None-KeyMeshing OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms keyMeshing(14) none(0) }
Gost28147-89-KeyMeshingAlgorithms ALGORITHM-IDENTIFIER ::= {
    { NULL IDENTIFIED BY id-Gost28147-89-CryptoPro-KeyMeshing } |
    { NULL IDENTIFIED BY id-Gost28147-89-None-KeyMeshing }
}
END -- Gost28147-89-ParamSetSyntax

```

10.4. GostR3411-94-DigestSyntax

```

GostR3411-94-DigestSyntax
  { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
    other(1) modules(1) gostR3411-94-DigestSyntax(1) 1 }
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for
-- use in the other ASN.1 modules contained within the Russian
-- Cryptography "GOST" & "GOST R" Specifications, and for the use
-- of other applications that will use them to access Russian
-- Cryptography services. Other applications may use them for
-- their own purposes, but this will not constrain extensions and
-- modifications needed to maintain or improve the Russian
-- Cryptography service.
  IMPORTS
    id-CryptoPro-algorithms, id-CryptoPro-hashes,
    ALGORITHM-IDENTIFIER,
    cryptographic-Gost-Useful-Definitions
  FROM Cryptographic-Gost-Useful-Definitions
    { iso(1) member-body(2) ru(643) rans(2)
      cryptopro(2) other(1) modules(1)
      cryptographic-Gost-Useful-Definitions(0) 1 }
  ;
-- GOST R 34.11-94 OID
  id-GostR3411-94 OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms gostR3411-94(9) }
-- GOST R 34.11-94 cryptographic parameter set OIDs
  id-GostR3411-94-TestParamSet OBJECT IDENTIFIER ::=
    { id-CryptoPro-hashes test(0) }
  id-GostR3411-94-CryptoProParamSet OBJECT IDENTIFIER ::=
    { id-CryptoPro-hashes cryptopro(1) }
-- GOST R 34.11-94 data types
  GostR3411-94-Digest ::= OCTET STRING (SIZE (32))
-- GOST R 34.11-94 digest algorithm & parameters
  GostR3411-94-DigestParameters ::=
    OBJECT IDENTIFIER (
      id-GostR3411-94-TestParamSet |
      -- Only for testing purposes
      id-GostR3411-94-CryptoProParamSet
    )
  GostR3411-94-DigestAlgorithms ALGORITHM-IDENTIFIER ::= {
    { NULL IDENTIFIED BY id-GostR3411-94 } |
    -- Assume id-GostR3411-94-CryptoProParamSet
    { GostR3411-94-DigestParameters
      IDENTIFIED BY id-GostR3411-94 }
  }
}

```

END -- GostR3411-94-DigestSyntax

10.5. GostR3411-94-ParamSetSyntax

```
GostR3411-94-ParamSetSyntax
  { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
    other(1) modules(1) gostR3411-94-ParamSetSyntax(7) 1 }
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for
-- use in the other ASN.1 modules contained within the Russian
-- Cryptography "GOST" & "GOST R" Specifications, and for the use
-- of other applications that will use them to access Russian
-- Cryptography services. Other applications may use them for
-- their own purposes, but this will not constrain extensions and
-- modifications needed to maintain or improve the Russian
-- Cryptography service.
IMPORTS
  gost28147-89-EncryptionSyntax,
  gostR3411-94-DigestSyntax,
  ALGORITHM-IDENTIFIER
FROM Cryptographic-Gost-Useful-Definitions
  { iso(1) member-body(2) ru(643) rans(2)
    cryptopro(2) other(1) modules(1)
    cryptographic-Gost-Useful-Definitions(0) 1 }
Gost28147-89-UZ
FROM Gost28147-89-EncryptionSyntax
  gost28147-89-EncryptionSyntax
id-GostR3411-94-TestParamSet,
id-GostR3411-94-CryptoProParamSet,
GostR3411-94-Digest
FROM GostR3411-94-DigestSyntax
  gostR3411-94-DigestSyntax
;
-- GOST R 34.11-94 cryptographic parameter sets:
-- OIDs for parameter sets are imported from
-- GostR3411-94-DigestSyntax
GostR3411-94-ParamSetParameters ::=
  SEQUENCE {
    hUZ Gost28147-89-UZ,    -- S-Box for digest
    h0  GostR3411-94-Digest -- initial digest value
  }
GostR3411-94-ParamSetAlgorithms ALGORITHM-IDENTIFIER ::= {
  { GostR3411-94-ParamSetParameters IDENTIFIED BY
    id-GostR3411-94-TestParamSet
  } |
  { GostR3411-94-ParamSetParameters IDENTIFIED BY
```

```

        id-GostR3411-94-CryptoProParamSet
    }
}
END -- GostR3411-94-ParamSetSyntax

```

10.6. GostR3410-94-PKISyntax

```

GostR3410-94-PKISyntax
{ iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
  other(1) modules(1) gostR3410-94-PKISyntax(2) 1 }
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for
-- use in the other ASN.1 modules contained within the Russian
-- Cryptography "GOST" & "GOST R" Specifications, and for the use
-- of other applications that will use them to access Russian
-- Cryptography services. Other applications may use them for
-- their own purposes, but this will not constrain extensions and
-- modifications needed to maintain or improve the Russian
-- Cryptography service.
  IMPORTS
    id-CryptoPro-algorithms,
    id-CryptoPro-signs, id-CryptoPro-exchanges,
    gost28147-89-EncryptionSyntax,
    gostR3411-94-DigestSyntax, ALGORITHM-IDENTIFIER,
    cryptographic-Gost-Useful-Definitions
  FROM Cryptographic-Gost-Useful-Definitions
    { iso(1) member-body(2) ru(643) rans(2)
      cryptopro(2) other(1) modules(1)
      cryptographic-Gost-Useful-Definitions(0) 1 }
  Gost28147-89-ParamSet
  FROM Gost28147-89-EncryptionSyntax
    gost28147-89-EncryptionSyntax
  id-GostR3411-94-TestParamSet,
  id-GostR3411-94-CryptoProParamSet
  FROM GostR3411-94-DigestSyntax gostR3411-94-DigestSyntax
;
-- GOST R 34.10-94 OIDs
id-GostR3410-94 OBJECT IDENTIFIER ::=
  { id-CryptoPro-algorithms gostR3410-94(20) }
id-GostR3410-94DH OBJECT IDENTIFIER ::=
  { id-CryptoPro-algorithms gostR3410-94DH(99) }
id-GostR3411-94-with-GostR3410-94 OBJECT IDENTIFIER ::=
  { id-CryptoPro-algorithms
    gostR3411-94-with-gostR3410-94(4) }
-- GOST R 34.10-94 public key parameter set OIDs
id-GostR3410-94-TestParamSet OBJECT IDENTIFIER ::=

```

```

    { id-CryptoPro-signs test(0) }
id-GostR3410-94-CryptoPro-A-ParamSet OBJECT IDENTIFIER ::=
    { id-CryptoPro-signs cryptopro-A(2) }
id-GostR3410-94-CryptoPro-B-ParamSet OBJECT IDENTIFIER ::=
    { id-CryptoPro-signs cryptopro-B(3) }
id-GostR3410-94-CryptoPro-C-ParamSet OBJECT IDENTIFIER ::=
    { id-CryptoPro-signs cryptopro-C(4) }
id-GostR3410-94-CryptoPro-D-ParamSet OBJECT IDENTIFIER ::=
    { id-CryptoPro-signs cryptopro-D(5) }
id-GostR3410-94-CryptoPro-XchA-ParamSet OBJECT IDENTIFIER ::=
    { id-CryptoPro-exchanges cryptopro-XchA(1) }
id-GostR3410-94-CryptoPro-XchB-ParamSet OBJECT IDENTIFIER ::=
    { id-CryptoPro-exchanges cryptopro-XchB(2) }
id-GostR3410-94-CryptoPro-XchC-ParamSet OBJECT IDENTIFIER ::=
    { id-CryptoPro-exchanges cryptopro-XchC(3) }
-- GOST R 34.10-94 data types
GostR3410-94-CertificateSignature ::=
    BIT STRING ( SIZE(256..512) )
GostR3410-94-PublicKey ::=
    OCTET STRING ( SIZE(
        64 |      -- Only for testing purposes
        128
    ) )
GostR3410-94-PublicKeyParameters ::=
    SEQUENCE {
        publicKeyParamSet
            OBJECT IDENTIFIER (
                id-GostR3410-94-TestParamSet |
                -- Only for testing purposes
                id-GostR3410-94-CryptoPro-A-ParamSet |
                id-GostR3410-94-CryptoPro-B-ParamSet |
                id-GostR3410-94-CryptoPro-C-ParamSet |
                id-GostR3410-94-CryptoPro-D-ParamSet |
                id-GostR3410-94-CryptoPro-XchA-ParamSet |
                id-GostR3410-94-CryptoPro-XchB-ParamSet |
                id-GostR3410-94-CryptoPro-XchC-ParamSet
            ),
        digestParamSet
            OBJECT IDENTIFIER (
                id-GostR3411-94-TestParamSet |
                -- Only for testing purposes
                id-GostR3411-94-CryptoProParamSet
            ),
        encryptionParamSet Gost28147-89-ParamSet OPTIONAL
    }
GostR3410-94-PublicKeyAlgorithms ALGORITHM-IDENTIFIER ::= {
    { GostR3410-94-PublicKeyParameters IDENTIFIED BY
        id-GostR3410-94 }

```



```

}
END -- GostR3410-94-PKISyntax

```

10.7. GostR3410-94-ParamSetSyntax

```

GostR3410-94-ParamSetSyntax
  { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
    other(1) modules(1) gostR3410-94-ParamSetSyntax(8) 1 }
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for
-- use in the other ASN.1 modules contained within the Russian
-- Cryptography "GOST" & "GOST R" Specifications, and for the use
-- of other applications that will use them to access Russian
-- Cryptography services. Other applications may use them for
-- their own purposes, but this will not constrain extensions and
-- modifications needed to maintain or improve the Russian
-- Cryptography service.
IMPORTS
  id-CryptoPro-algorithms,
  id-CryptoPro-signs, id-CryptoPro-exchanges,
  gostR3410-94-PKISyntax, ALGORITHM-IDENTIFIER,
  cryptographic-Gost-Useful-Definitions
FROM Cryptographic-Gost-Useful-Definitions
  { iso(1) member-body(2) ru(643) rans(2)
    cryptopro(2) other(1) modules(1)
    cryptographic-Gost-Useful-Definitions(0) 1 }
  id-GostR3410-94,
  id-GostR3410-94-TestParamSet,
  id-GostR3410-94-CryptoPro-A-ParamSet,
  id-GostR3410-94-CryptoPro-B-ParamSet,
  id-GostR3410-94-CryptoPro-C-ParamSet,
  id-GostR3410-94-CryptoPro-D-ParamSet,
  id-GostR3410-94-CryptoPro-XchA-ParamSet,
  id-GostR3410-94-CryptoPro-XchB-ParamSet,
  id-GostR3410-94-CryptoPro-XchC-ParamSet
FROM GostR3410-94-PKISyntax gostR3410-94-PKISyntax
AlgorithmIdentifier
FROM PKIX1Explicit88 {iso(1) identified-organization(3)
dod(6) internet(1) security(5) mechanisms(5) pkix(7)
id-mod(0) id-pkix1-explicit-88(1)}
;
-- GOST R 34.10-94 public key parameter sets:
-- OIDs for parameter sets are imported from
-- GostR3410-94-PKISyntax
GostR3410-94-ParamSetParameters-t ::= INTEGER (512 | 1024)
-- 512 - only for testing purposes

```

```

GostR3410-94-ParamSetParameters ::=
  SEQUENCE {
    t GostR3410-94-ParamSetParameters-t,
    p INTEGER, --  $2^{1020} < p < 2^{1024}$  or  $2^{509} < p < 2^{512}$ 
    q INTEGER, --  $2^{254} < q < 2^{256}$ 
    a INTEGER, --  $1 < a < p-1 < 2^{1024}-1$ 
    validationAlgorithm
      AlgorithmIdentifier OPTIONAL
      -- {{ GostR3410-94-ValidationAlgorithms }}
  }
GostR3410-94-ParamSetAlgorithm ALGORITHM-IDENTIFIER ::= {
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-TestParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-A-ParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-B-ParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-C-ParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-D-ParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-XchA-ParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-XchB-ParamSet } |
  { GostR3410-94-ParamSetParameters IDENTIFIED BY
    id-GostR3410-94-CryptoPro-XchC-ParamSet }
}
-- GOST R 34.10-94 validation/constructor
id-GostR3410-94-a OBJECT IDENTIFIER ::=
  { id-GostR3410-94 a(1) }
id-GostR3410-94-aBis OBJECT IDENTIFIER ::=
  { id-GostR3410-94 aBis(2) }
id-GostR3410-94-b OBJECT IDENTIFIER ::=
  { id-GostR3410-94 b(3) }
id-GostR3410-94-bBis OBJECT IDENTIFIER ::=
  { id-GostR3410-94 bBis(4) }
GostR3410-94-ValidationParameters-c ::=
  INTEGER (0 .. 65535)
GostR3410-94-ValidationParameters ::=
  SEQUENCE {
    x0 GostR3410-94-ValidationParameters-c,
    c GostR3410-94-ValidationParameters-c,
    d INTEGER OPTIONAL --  $1 < d < p-1 < 2^{1024}-1$ 
  }
GostR3410-94-ValidationBisParameters-c ::=
  INTEGER (0 .. 4294967295)

```

```

GostR3410-94-ValidationBisParameters ::=
  SEQUENCE {
    x0  GostR3410-94-ValidationBisParameters-c,
    c    GostR3410-94-ValidationBisParameters-c,
    d    INTEGER OPTIONAL -- 1 < d < p-1 < 2^1024-1
  }
GostR3410-94-ValidationAlgorithms ALGORITHM-IDENTIFIER ::= {
  { GostR3410-94-ValidationParameters IDENTIFIED BY
    id-GostR3410-94-a } |
  { GostR3410-94-ValidationBisParameters IDENTIFIED BY
    id-GostR3410-94-aBis } |
  { GostR3410-94-ValidationParameters IDENTIFIED BY
    id-GostR3410-94-b } |
  { GostR3410-94-ValidationBisParameters IDENTIFIED BY
    id-GostR3410-94-bBis }
}
END -- GostR3410-94-ParamSetSyntax

```

10.8. GostR3410-2001-PKISyntax

```

GostR3410-2001-PKISyntax
  { iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
    other(1) modules(1) gostR3410-2001-PKISyntax(9) 1 }
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for
-- use in the other ASN.1 modules contained within the Russian
-- Cryptography "GOST" & "GOST R" Specifications, and for the use
-- of other applications that will use them to access Russian
-- Cryptography services. Other applications may use them for
-- their own purposes, but this will not constrain extensions and
-- modifications needed to maintain or improve the Russian
-- Cryptography service.
IMPORTS
  id-CryptoPro-algorithms,
  id-CryptoPro-ecc-signs, id-CryptoPro-ecc-exchanges,
  gost28147-89-EncryptionSyntax,
  gostR3411-94-DigestSyntax, ALGORITHM-IDENTIFIER,
  cryptographic-Gost-Useful-Definitions
FROM Cryptographic-Gost-Useful-Definitions
  { iso(1) member-body(2) ru(643) rans(2)
    cryptopro(2) other(1) modules(1)
    cryptographic-Gost-Useful-Definitions(0) 1 }
Gost28147-89-ParamSet
FROM Gost28147-89-EncryptionSyntax
  gost28147-89-EncryptionSyntax

```

```

    id-GostR3411-94-TestParamSet,
    id-GostR3411-94-CryptoProParamSet
    FROM GostR3411-94-DigestSyntax gostR3411-94-DigestSyntax
;
-- GOST R 34.10-2001 OIDs
id-GostR3410-2001 OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms gostR3410-2001(19) }
id-GostR3410-2001DH OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms gostR3410-2001DH(98) }
id-GostR3411-94-with-GostR3410-2001 OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms
      gostR3411-94-with-gostR3410-2001(3) }
-- GOST R 34.10-2001 public key parameter set OIDs
id-GostR3410-2001-TestParamSet OBJECT IDENTIFIER ::=
    { id-CryptoPro-ecc-signs test(0) }
id-GostR3410-2001-CryptoPro-A-ParamSet OBJECT IDENTIFIER ::=
    { id-CryptoPro-ecc-signs cryptopro-A(1) }
id-GostR3410-2001-CryptoPro-B-ParamSet OBJECT IDENTIFIER ::=
    { id-CryptoPro-ecc-signs cryptopro-B(2) }
id-GostR3410-2001-CryptoPro-C-ParamSet OBJECT IDENTIFIER ::=
    { id-CryptoPro-ecc-signs cryptopro-C(3) }
id-GostR3410-2001-CryptoPro-XchA-ParamSet
    OBJECT IDENTIFIER ::=
        { id-CryptoPro-ecc-exchanges cryptopro-XchA(0) }
id-GostR3410-2001-CryptoPro-XchB-ParamSet
    OBJECT IDENTIFIER ::=
        { id-CryptoPro-ecc-exchanges cryptopro-XchB(1) }
-- GOST R 34.10-2001 Data Types
GostR3410-2001-CertificateSignature ::=
    BIT STRING ( SIZE(256..512) )
GostR3410-2001-PublicKey ::=
    OCTET STRING ( SIZE(64) )
GostR3410-2001-PublicKeyParameters ::=
    SEQUENCE {
        publicKeyParamSet
            OBJECT IDENTIFIER (
                id-GostR3410-2001-TestParamSet |
                -- Only for testing purposes
                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
            ),
        digestParamSet
            OBJECT IDENTIFIER (
                id-GostR3411-94-TestParamSet |
                -- Only for testing purposes

```

```

        id-GostR3411-94-CryptoProParamSet
    ),
    encryptionParamSet Gost28147-89-ParamSet OPTIONAL
}
GostR3410-2001-PublicKeyAlgorithms ALGORITHM-IDENTIFIER ::= {
    { GostR3410-2001-PublicKeyParameters IDENTIFIED BY
        id-GostR3410-2001 }
}
END -- GostR3410-2001-PKISyntax

```

10.9. GostR3410-2001-ParamSetSyntax

```

GostR3410-2001-ParamSetSyntax
{ iso(1) member-body(2) ru(643) rans(2) cryptopro(2)
  other(1) modules(1) gostR3410-2001-ParamSetSyntax(12) 1 }
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for
-- use in the other ASN.1 modules contained within the Russian
-- Cryptography "GOST" & "GOST R" Specifications, and for the use
-- of other applications that will use them to access Russian
-- Cryptography services. Other applications may use them for
-- their own purposes, but this will not constrain extensions and
-- modifications needed to maintain or improve the Russian
-- Cryptography service.
IMPORTS
    gostR3410-2001-PKISyntax, ALGORITHM-IDENTIFIER,
    cryptographic-Gost-Useful-Definitions
FROM Cryptographic-Gost-Useful-Definitions
    { iso(1) member-body(2) ru(643) rans(2)
      cryptopro(2) other(1) modules(1)
        cryptographic-Gost-Useful-Definitions(0) 1 }
id-GostR3410-2001,
id-GostR3410-2001-TestParamSet,
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
FROM GostR3410-2001-PKISyntax gostR3410-2001-PKISyntax
;
GostR3410-2001-ParamSetParameters ::=
SEQUENCE {
    a INTEGER, -- 0 < a < p < 2^256
    b INTEGER, -- 0 < b < p < 2^256
    p INTEGER, -- 2^254 < p < 2^256
    q INTEGER, -- 2^254 < q < 2^256
}

```

```

        x INTEGER, -- 0 < x < p < 2^256
        y INTEGER  -- 0 < y < p < 2^256
    }
-- GOST R 34.10-2001 public key parameter set:
-- OIDs for parameter sets are imported from
-- GostR3410-2001-PKISyntax
GostR3410-2001-ParamSetAlgorithm ALGORITHM-IDENTIFIER ::= {
    { GostR3410-2001-ParamSetParameters IDENTIFIED BY
      id-GostR3410-2001-TestParamSet } |
    { GostR3410-2001-ParamSetParameters IDENTIFIED BY
      id-GostR3410-2001-CryptoPro-A-ParamSet } |
    { GostR3410-2001-ParamSetParameters IDENTIFIED BY
      id-GostR3410-2001-CryptoPro-B-ParamSet } |
    { GostR3410-2001-ParamSetParameters IDENTIFIED BY
      id-GostR3410-2001-CryptoPro-C-ParamSet } |
    { GostR3410-2001-ParamSetParameters IDENTIFIED BY
      id-GostR3410-2001-CryptoPro-XchA-ParamSet } |
    { GostR3410-2001-ParamSetParameters IDENTIFIED BY
      id-GostR3410-2001-CryptoPro-XchB-ParamSet }
}
END -- GostR3410-2001-ParamSetSyntax

```

11. Appendix Parameters

Parameters here are given as SEQUENCE OF AlgorithmIdentifier in ASN.1 DER encoding [X.660], stored in the same format as the examples in [RFC4134], can be extracted using the same program.

If you want to extract without the program, copy all the lines between the "|>" and "|<" markers, remove any page breaks, and remove the "|" in the first column of each line. The result is a valid Base64 blob that can be processed by any Base64 decoder.

11.1. Encryption Algorithm Parameters

For each AlgorithmIdentifier in this sequence, the parameters field contains Gost28147-89-ParamSetParameters.

```

0 30 480: SEQUENCE {
4 30 94: SEQUENCE {
6 06 7: OBJECT IDENTIFIER
: id-Gost28147-89-TestParamSet
15 30 83: SEQUENCE {
17 04 64: OCTET STRING
: 4C DE 38 9C 29 89 EF B6 FF EB 56 C5 5E C2 9B 02
: 98 75 61 3B 11 3F 89 60 03 97 0C 79 8A A1 D5 5D
: E2 10 AD 43 37 5D B3 8E B4 2C 77 E7 CD 46 CA FA
: D6 6A 20 1F 70 F4 1E A4 AB 03 F2 21 65 B8 44 D8

```

```

83 02 1:    INTEGER 0
86 02 1:    INTEGER 64
89 30 9:    SEQUENCE {
91 06 7:    OBJECT IDENTIFIER
           :      id-Gost28147-89-None-KeyMeshing
           :    }
           :  }
           : }
100 30 94:  SEQUENCE {
102 06 7:    OBJECT IDENTIFIER
           :      id-Gost28147-89-CryptoPro-A-ParamSet
111 30 83:  SEQUENCE {
113 04 64:  OCTET STRING

           -- K1 K2 K3 K4 K5 K6 K7 K8
           -- 9 3 E E B 3 1 B
           -- 6 7 4 7 5 A D A
           -- 3 E 6 A 1 D 2 F
           -- 2 9 2 C 9 C 9 5
           -- 8 8 B D 8 1 7 0
           -- B A 3 1 D 2 A C
           -- 1 F D 3 F 0 6 E
           -- 7 0 8 9 0 B 0 8
           -- A 5 C 0 E 7 8 6
           -- 4 2 F 2 4 5 C 2
           -- E 6 5 B 2 9 4 3
           -- F C A 4 3 4 5 9
           -- C B 0 F C 8 F 1
           -- 0 4 7 8 7 F 3 7
           -- D D 1 5 A E B D
           -- 5 1 9 6 6 6 E 4

           :    93 EE B3 1B 67 47 5A DA 3E 6A 1D 2F 29 2C 9C 95
           :    88 BD 81 70 BA 31 D2 AC 1F D3 F0 6E 70 89 0B 08
           :    A5 C0 E7 86 42 F2 45 C2 E6 5B 29 43 FC A4 34 59
           :    CB 0F C8 F1 04 78 7F 37 DD 15 AE BD 51 96 66 E4
179 02 1:    INTEGER 1
182 02 1:    INTEGER 64
185 30 9:    SEQUENCE {
187 06 7:    OBJECT IDENTIFIER
           :      id-Gost28147-89-CryptoPro-KeyMeshing
           :    }
           :  }
           : }
196 30 94:  SEQUENCE {
198 06 7:    OBJECT IDENTIFIER
           :      id-Gost28147-89-CryptoPro-B-ParamSet
207 30 83:  SEQUENCE {

```

```

209 04 64:  OCTET STRING
      :    80 E7 28 50 41 C5 73 24 B2 00 C2 AB 1A AD F6 BE
      :    34 9B 94 98 5D 26 5D 13 05 D1 AE C7 9C B2 BB 31
      :    29 73 1C 7A E7 5A 41 42 A3 8C 07 D9 CF FF DF 06
      :    DB 34 6A 6F 68 6E 80 FD 76 19 E9 85 FE 48 35 EC
275 02 1:   INTEGER 1
278 02 1:   INTEGER 64
281 30 9:   SEQUENCE {
283 06 7:   OBJECT IDENTIFIER
      :     id-Gost28147-89-CryptoPro-KeyMeshing
      :   }
      : }
      : }
292 30 94:  SEQUENCE {
294 06 7:   OBJECT IDENTIFIER
      :     id-Gost28147-89-CryptoPro-C-ParamSet
303 30 83:  SEQUENCE {
305 04 64:  OCTET STRING
      :    10 83 8C A7 B1 26 D9 94 C7 50 BB 60 2D 01 01 85
      :    9B 45 48 DA D4 9D 5E E2 05 FA 12 2F F2 A8 24 0E
      :    48 3B 97 FC 5E 72 33 36 8F C9 C6 51 EC D7 E5 BB
      :    A9 6E 6A 4D 7A EF F0 19 66 1C AF C3 33 B4 7D 78
371 02 1:   INTEGER 1
374 02 1:   INTEGER 64
377 30 9:   SEQUENCE {
379 06 7:   OBJECT IDENTIFIER
      :     id-Gost28147-89-CryptoPro-KeyMeshing
      :   }
      : }
      : }
388 30 94:  SEQUENCE {
390 06 7:   OBJECT IDENTIFIER
      :     id-Gost28147-89-CryptoPro-D-ParamSet
399 30 83:  SEQUENCE {
401 04 64:  OCTET STRING
      :    FB 11 08 31 C6 C5 C0 0A 23 BE 8F 66 A4 0C 93 F8
      :    6C FA D2 1F 4F E7 25 EB 5E 60 AE 90 02 5D BB 24
      :    77 A6 71 DC 9D D2 3A 83 E8 4B 64 C5 D0 84 57 49
      :    15 99 4C B7 BA 33 E9 AD 89 7F FD 52 31 28 16 7E
467 02 1:   INTEGER 1
470 02 1:   INTEGER 64
473 30 9:   SEQUENCE {
475 06 7:   OBJECT IDENTIFIER
      :     id-Gost28147-89-CryptoPro-KeyMeshing
      :   }
      : }
      : }
      : }

```



```

|>Gost28147-89-ParamSetParameters.bin
|MIIB4DBeBgcqhQMCAh8AMFMEQEzeOJwpie+2/+tWxV7CmwKYdWE7ET+JYAOXDhMk
|odVd4hCtQzdds460LHfnzUbK+tZqIB9w9B6kqwPyIWW4RNgCAQACAUAwCQYHkoUD
|AgIOADBeBgcqhQMCAh8BMFMEQJPusxtnr1raPmodLyksnJWivYFwu jHSrB/T8G5w
|iQsIpcDnhkLyRcLmWylD/KQ0WcsPyPEEeH833RWuvVGWZuQCAQECAUAWCQYHkoUD
|AgIOATBeBgcqhQMCAh8CMFMEQIDnKFBBxXMksgDCqxt9r40m5SYXSZdEwXRrsec
|srsxKXMceudaQUKjjAfZz//fBts0am9oboD9dhnphf5INewCAQECAUAWCQYHkoUD
|AgIOATBeBgcqhQMCAh8DMFMEQBCDjKexJtmUxlC7YC0BAYWbRUja1Jle4gX6Ei/y
|qCQOSDuX/F5yMzaPycZR7Nflu6luak167/AZZhyvwz00fXgCAQECAUAWCQYHkoUD
|AgIOATBeBgcqhQMCAh8EMFMEQPsRCDHGxcAKI76PZqQMk/hs+tIfT+cl615grpAC
|Xbskd6Zx3J3S0oPoS2TF0IRXSRWZTLe6M+mtiX/9UjEoFn4CAQECAUAWCQYHkoUD
|AgIOAQ==
|<Gost28147-89-ParamSetParameters.bin

```

11.2. Digest Algorithm Parameters

For each AlgorithmIdentifier in this sequence, the parameters field contains GostR3411-94-ParamSetParameters.

```

0 30 226: SEQUENCE {
3 30 111: SEQUENCE {
5 06 7: OBJECT IDENTIFIER
: id-GostR3411-94-TestParamSet
14 30 100: SEQUENCE {
16 04 64: OCTET STRING

--      pi1 pi2 pi3 pi4 pi5 pi6 pi7 pi8
--      4   E   5   7   6   4   D   1
--      A   B   8   D   C   B   B   F
--      9   4   1   A   7   A   4   D
--      2   C   D   1   1   0   1   0
--      D   6   A   0   5   7   3   5
--      8   D   3   8   F   2   F   7
--      0   F   4   9   D   1   5   A
--      E   A   2   F   8   D   9   4
--      6   2   E   E   4   3   0   9
--      B   3   F   4   A   6   A   2
--      1   8   C   6   9   8   E   3
--      C   1   7   C   E   5   7   E
--      7   0   6   B   0   9   6   6
--      F   7   0   2   3   C   8   B
--      5   5   9   5   B   F   2   8
--      3   9   B   3   2   E   C   C

:      4E 57 64 D1 AB 8D CB BF 94 1A 7A 4D 2C D1 10 10
:      D6 A0 57 35 8D 38 F2 F7 0F 49 D1 5A EA 2F 8D 94
:      62 EE 43 09 B3 F4 A6 A2 18 C6 98 E3 C1 7C E5 7E
:      70 6B 09 66 F7 02 3C 8B 55 95 BF 28 39 B3 2E CC

```

```

82 04 32: OCTET STRING
      : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      : }
      : }
116 30 111: SEQUENCE {
118 06 7: OBJECT IDENTIFIER
      : id-GostR3411-94-CryptoProParamSet
127 30 100: SEQUENCE {
129 04 64: OCTET STRING
      : A5 74 77 D1 4F FA 66 E3 54 C7 42 4A 60 EC B4 19
      : 82 90 9D 75 1D 4F C9 0B 3B 12 2F 54 79 08 A0 AF
      : D1 3E 1A 38 C7 B1 81 C6 E6 56 05 87 03 25 EB FE
      : 9C 6D F8 6D 2E AB DE 20 BA 89 3C 92 F8 D3 53 BC
195 04 32: OCTET STRING
      : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      : }
      : }
      : }

```

```

|>GostR3411-94-ParamSetParameters.bin
|MIHiMG8GByqFAwICHgAwZARATldk0auNy7+UGnpNLNEQENagVzWNOPL3D0nRWuov
|jZRi7kMJs/SmohjGmOPBfOV+cGsJZvcCPItVlb8oObMuzAQgAAAAAAAAAAAAAAAA
|AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAwbwYHKoUDAgIeATBkBECldHfRT/pm41THQkpg
|7LQZgpCddR1PyQs7Ei9UeQigr9E+GjjHsYHG5lYFhwMl6/6cbfhtLqveILqJPJL4
|0l08BCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==
|<GostR3411-94-ParamSetParameters.bin

```

11.3. GOST R 34.10-94 Public Key Algorithm Parameters

For each AlgorithmIdentifier in this sequence, the parameters field contains GostR3410-94-ParamSetParameters.

```

0 30 2882: SEQUENCE {
4 30 209: SEQUENCE {
7 06 7: OBJECT IDENTIFIER
      : id-GostR3410-94-TestParamSet
16 30 197: SEQUENCE {
19 02 2: INTEGER 512
23 02 65: INTEGER
      : 00 EE 81 72 AE 89 96 60 8F B6 93 59 B8 9E B8 2A
      : 69 85 45 10 E2 97 7A 4D 63 BC 97 32 2C E5 DC 33
      : 86 EA 0A 12 B3 43 E9 19 0F 23 17 75 39 84 58 39
      : 78 6B B0 C3 45 D1 65 97 6E F2 19 5E C9 B1 C3 79
      : E3
90 02 33: INTEGER
      : 00 98 91 5E 7E C8 26 5E DF CD A3 1E 88 F2 48 09

```

```

      :      DD B0 64 BD C7 28 5D D5 0D 72 89 F0 AC 6F 49 DD
      :      2D
125 02 65:  INTEGER
      :      00 9E 96 03 15 00 C8 77 4A 86 95 82 D4 AF DE 21
      :      27 AF AD 25 38 B4 B6 27 0A 6F 7C 88 37 B5 0D 50
      :      F2 06 75 59 84 A4 9E 50 93 04 D6 48 BE 2A B5 AA
      :      B1 8E BE 2C D4 6A C3 D8 49 5B 14 2A A6 CE 23 E2
      :      1C
192 30 22:  SEQUENCE {
194 06  7:  OBJECT IDENTIFIER id-GostR3410-94-a
203 30 11:  SEQUENCE {
205 02  2:  INTEGER 24265
209 02  2:  INTEGER 29505
213 02  1:  INTEGER 2
      :      }
      :      }
      :      }
216 30 342: SEQUENCE {
220 06  7:  OBJECT IDENTIFIER
      :      id-GostR3410-94-CryptoPro-A-ParamSet
229 30 329: SEQUENCE {
233 02  2:  INTEGER 1024
237 02 129:  INTEGER
      :      00 B4 E2 5E FB 01 8E 3C 8B 87 50 5E 2A 67 55 3C
      :      5E DC 56 C2 91 4B 7E 4F 89 D2 3F 03 F0 33 77 E7
      :      0A 29 03 48 9D D6 0E 78 41 8D 3D 85 1E DB 53 17
      :      C4 87 1E 40 B0 42 28 C3 B7 90 29 63 C4 B7 D8 5D
      :      52 B9 AA 88 F2 AF DB EB 28 DA 88 69 D6 DF 84 6A
      :      1D 98 92 4E 92 55 61 BD 69 30 0B 9D DD 05 D2 47
      :      B5 92 2D 96 7C BB 02 67 18 81 C5 7D 10 E5 EF 72
      :      D3 E6 DA D4 22 3D C8 2A A1 F7 D0 29 46 51 A4 80
      :      DF
369 02 33:  INTEGER
      :      00 97 24 32 A4 37 17 8B 30 BD 96 19 5B 77 37 89
      :      AB 2F FF 15 59 4B 17 6D D1 75 B6 32 56 EE 5A F2
      :      CF
404 02 129:  INTEGER
      :      00 8F D3 67 31 23 76 54 BB E4 1F 5F 1F 84 53 E7
      :      1C A4 14 FF C2 2C 25 D9 15 30 9E 5D 2E 62 A2 A2
      :      6C 71 11 F3 FC 79 56 8D AF A0 28 04 2F E1 A5 2A
      :      04 89 80 5C 0D E9 A1 A4 69 C8 44 C7 CA BB EE 62
      :      5C 30 78 88 8C 1D 85 EE A8 83 F1 AD 5B C4 E6 77
      :      6E 8E 1A 07 50 91 2D F6 4F 79 95 64 99 F1 E1 82
      :      47 5B 0B 60 E2 63 2A DC D8 CF 94 E9 C5 4F D1 F3
      :      B1 09 D8 1F 00 BF 2A B8 CB 86 2A DF 7D 40 B9 36
      :      9A
536 30 24:  SEQUENCE {

```

```

538 06 7: OBJECT IDENTIFIER id-GostR3410-94-bBis
547 30 13: SEQUENCE {
549 02 4: INTEGER 1376285941
555 02 5: INTEGER
: 00 EE 39 AD B3
: }
: }
: }
: }
562 30 427: SEQUENCE {
566 06 7: OBJECT IDENTIFIER
: id-GostR3410-94-CryptoPro-B-ParamSet
575 30 414: SEQUENCE {
579 02 2: INTEGER 1024
583 02 129: INTEGER
: 00 C6 97 1F C5 75 24 B3 0C 90 18 C5 E6 21 DE 15
: 49 97 36 85 4F 56 A6 F8 AE E6 5A 7A 40 46 32 B1
: BC F0 34 9F FC AF CB 0A 10 31 77 97 1F C1 61 2A
: DC DB 8C 8C C9 38 C7 02 25 C8 FD 12 AF F0 1B 1D
: 06 4E 0A D6 FD E6 AB 91 59 16 6C B9 F2 FC 17 1D
: 92 F0 CC 7B 6A 6B 2C D7 FA 34 2A CB E2 C9 31 5A
: 42 D5 76 B1 EC CE 77 A9 63 15 7F 3D 0B D9 6A 8E
: B0 B0 F3 50 2A D2 38 10 1B 05 11 63 34 F1 E5 B7
: AB
715 02 33: INTEGER
: 00 B0 9D 63 4C 10 89 9C D7 D4 C3 A7 65 74 03 E0
: 58 10 B0 7C 61 A6 88 BA B2 C3 7F 47 5E 30 8B 06
: 07
750 02 128: INTEGER
: 3D 26 B4 67 D9 4A 3F FC 9D 71 BF 8D B8 93 40 84
: 13 72 64 F3 C2 E9 EB 16 DC A2 14 B8 BC 7C 87 24
: 85 33 67 44 93 4F D2 EF 59 43 F9 ED 0B 74 5B 90
: AA 3E C8 D7 0C DC 91 68 24 78 B6 64 A2 E1 F8 FB
: 56 CE F2 97 2F EE 7E DB 08 4A F7 46 41 9B 85 4F
: AD 02 CC 3E 36 46 FF 2E 1A 18 DD 4B EB 3C 44 F7
: F2 74 55 88 02 96 49 67 45 46 CC 91 87 C2 07 FB
: 8F 2C EC E8 E2 29 3F 68 39 5C 47 04 AF 04 BA B5
881 30 110: SEQUENCE {
883 06 7: OBJECT IDENTIFIER id-GostR3410-94-bBis
892 30 99: SEQUENCE {
894 02 4: INTEGER 1536654555
900 02 4: INTEGER 1855361757
906 02 85: INTEGER
: 00 BC 3C BB DB 7E 6F 84 82 86 E1 9A D9 A2 7A 8E
: 29 7E 5B 71 C5 3D D9 74 CD F6 0F 93 73 56 DF 69
: CB C9 7A 30 0C CC 71 68 5C 55 30 46 14 7F 11 56
: 8C 4F DD F3 63 D9 D8 86 43 83 45 A6 2C 3B 75 96
: 3D 65 46 AD FA BF 31 B3 12 90 D1 2C AE 65 EC B8

```

```

      :      30 9E F6 67 82
      :      }
      :      }
      :      }
      :      }
993 30 351: SEQUENCE {
997 06 7:  OBJECT IDENTIFIER
      :      id-Gostr3410-94-CryptoPro-C-ParamSet
1006 30 338: SEQUENCE {
1010 02 2:  INTEGER 1024
1014 02 129: INTEGER
      :      00 9D 88 E6 D7 FE 33 13 BD 2E 74 5C 7C DD 2A B9
      :      EE 4A F3 C8 89 9E 84 7D E7 4A 33 78 3E A6 8B C3
      :      05 88 BA 1F 73 8C 6A AF 8A B3 50 53 1F 18 54 C3
      :      83 7C C3 C8 60 FF D7 E2 E1 06 C3 F6 3B 3D 8A 4C
      :      03 4C E7 39 42 A6 C3 D5 85 B5 99 CF 69 5E D7 A3
      :      C4 A9 3B 2B 94 7B 71 57 BB 1A 1C 04 3A B4 1E C8
      :      56 6C 61 45 E9 38 A6 11 90 6D E0 D3 2E 56 24 94
      :      56 9D 7E 99 9A 0D DA 5C 87 9B DD 91 FE 12 4D F1
      :      E9
1146 02 33: INTEGER
      :      00 FA DD 19 7A BD 19 A1 B4 65 3E EC F7 EC A4 D6
      :      A2 2B 1F 7F 89 3B 64 1F 90 16 41 FB B5 55 35 4F
      :      AF
1181 02 128: INTEGER
      :      74 47 ED 71 56 31 05 99 07 0B 12 60 99 47 A5 C8
      :      C8 A8 62 5C F1 CF 25 2B 40 7B 33 1F 93 D6 39 DD
      :      D1 BA 39 26 56 DE CA 99 2D D0 35 35 43 29 A1 E9
      :      5A 6E 32 D6 F4 78 82 D9 60 B8 F1 0A CA FF 79 6D
      :      13 CD 96 11 F8 53 DA B6 D2 62 34 83 E4 67 88 70
      :      84 93 93 7A 1A 29 44 25 98 AE C2 E0 74 20 22 56
      :      34 40 FE 9C 18 74 0E CE 67 65 AC 05 FA F0 24 A6
      :      4B 02 6E 7E 40 88 40 81 9E 96 2E 7E 5F 40 1A E3
1312 30 34: SEQUENCE {
1314 06 7:  OBJECT IDENTIFIER id-Gostr3410-94-bBis
1323 30 23: SEQUENCE {
1325 02 4:  INTEGER 1132758852
1331 02 5:  INTEGER
      :      00 B5 0A 82 6D
1338 02 8:  INTEGER
      :      7F 57 5E 81 94 BC 5B DF
      :      }
      :      }
      :      }
      :      }
1348 30 371: SEQUENCE {
1352 06 7:  OBJECT IDENTIFIER
      :      id-Gostr3410-94-CryptoPro-D-ParamSet

```

```

1361 30 358: SEQUENCE {
1365 02 2:   INTEGER 1024
1369 02 129:  INTEGER
:           00 80 F1 02 D3 2B 0F D1 67 D0 69 C2 7A 30 7A DA
:           D2 C4 66 09 19 04 DB AA 55 D5 B8 CC 70 26 F2 F7
:           A1 91 9B 89 0C B6 52 C4 0E 05 4E 1E 93 06 73 5B
:           43 D7 B2 79 ED DF 91 02 00 1C D9 E1 A8 31 FE 8A
:           16 3E ED 89 AB 07 CF 2A BE 82 42 AC 9D ED DD BF
:           98 D6 2C DD D1 EA 4F 5F 15 D3 A4 2A 66 77 BD D2
:           93 B2 42 60 C0 F2 7C 0F 1D 15 94 86 14 D5 67 B6
:           6F A9 02 BA A1 1A 69 AE 3B CE AD BB 83 E3 99 C9
:           B5
1501 02 33:  INTEGER
:           00 F0 F5 44 C4 18 AA C2 34 F6 83 F0 33 51 1B 65
:           C2 16 51 A6 07 8B DA 2D 69 BB 9F 73 28 67 50 21
:           49
1536 02 128:  INTEGER
:           6B CC 0B 4F AD B3 88 9C 1E 06 AD D2 3C C0 9B 8A
:           B6 EC DE DF 73 F0 46 32 59 5E E4 25 00 05 D6 AF
:           5F 5A DE 44 CB 1E 26 E6 26 3C 67 23 47 CF A2 6F
:           9E 93 93 68 1E 6B 75 97 33 78 4C DE 5D BD 9A 14
:           A3 93 69 DF D9 9F A8 5C C0 D1 02 41 C4 01 03 43
:           F3 4A 91 39 3A 70 6C F1 26 77 CB FA 1F 57 8D 6B
:           6C FB E8 A1 24 2C FC C9 4B 3B 65 3A 47 6E 14 5E
:           38 62 C1 8C C3 FE D8 25 7C FE F7 4C DB 20 5B F1
1667 30 54:  SEQUENCE {
1669 06 7:   OBJECT IDENTIFIER id-Gostr3410-94-bBis
1678 30 43:  SEQUENCE {
1680 02 4:   INTEGER 333089693
1686 02 5:   INTEGER
:           00 A0 E9 DE 4B
1693 02 28:  INTEGER
:           41 AB 97 85 7F 42 61 43 55 D3 2D B0 B1 06 9F 10
:           9A 4D A2 83 67 6C 7C 53 A6 81 85 B4
:           }
:       }
:   }
1723 30 396: SEQUENCE {
1727 06 7:   OBJECT IDENTIFIER
:           id-Gostr3410-94-CryptoPro-XchA-ParamSet
1736 30 383: SEQUENCE {
1740 02 2:   INTEGER 1024
1744 02 129:  INTEGER
:           00 CA 3B 3F 2E EE 9F D4 63 17 D4 95 95 A9 E7 51
:           8E 6C 63 D8 F4 EB 4D 22 D1 0D 28 AF 0B 88 39 F0
:           79 F8 28 9E 60 3B 03 53 07 84 B9 BB 5A 1E 76 85
:           9E 48 50 C6 70 C7 B7 1C 0D F8 4C A3 E0 D6 C1 77

```

```

:      FE 9F 78 A9 D8 43 32 30 A8 83 CD 82 A2 B2 B5 C7
:      A3 30 69 80 27 85 70 CD B7 9B F0 10 74 A6 9C 96
:      23 34 88 24 B0 C5 37 91 D5 3C 6A 78 CA B6 9E 1C
:      FB 28 36 86 11 A3 97 F5 0F 54 1E 16 DB 34 8D BE
:      5F
1876 02 33: INTEGER
:      00 CA E4 D8 5F 80 C1 47 70 4B 0C A4 8E 85 FB 00
:      A9 05 7A A4 AC C4 46 68 E1 7F 19 96 D7 15 26 90
:      D9
1911 02 129: INTEGER
:      00 BE 27 D6 52 F2 F1 E3 39 DA 73 42 11 B8 5B 06
:      AE 4D E2 36 AA 8F BE EB 3F 1A DC C5 2C D4 38 53
:      77 7E 83 4A 6A 51 81 38 67 8A 8A DB D3 A5 5C 70
:      A7 EA B1 BA 7A 07 19 54 86 77 AA F4 E6 09 FF B4
:      7F 6B 9D 7E 45 B0 D0 6D 83 D7 AD C5 33 10 AB D8
:      57 83 E7 31 7F 7E C7 32 68 B6 A9 C0 8D 26 0B 85
:      D8 48 56 96 CA 39 C1 7B 17 F0 44 D1 E0 50 48 90
:      36 AB D3 81 C5 E6 BF 82 BA 35 2A 1A FF 13 66 01
:      AF
2043 30 78: SEQUENCE {
2045 06 7:  OBJECT IDENTIFIER id-Gostr3410-94-bBis
2054 30 67:  SEQUENCE {
2056 02 5:    INTEGER
:          00 D0 5E 9F 14
2063 02 4:    INTEGER 1177570399
2069 02 52:    INTEGER
:          35 AB 87 53 99 CD A3 3C 14 6C A6 29 66 0E 5A 5E
:          5C 07 71 4C A3 26 DB 03 2D D6 75 19 95 CD B9 0A
:          61 2B 92 28 93 2D 83 02 70 4E C2 4A 5D EF 77 39
:          C5 81 3D 83
:      }
:  }
:  }
:  }
2123 30 375: SEQUENCE {
2127 06 7:  OBJECT IDENTIFIER
:          id-Gostr3410-94-CryptoPro-XchB-ParamSet
2136 30 362: SEQUENCE {
2140 02 2:  INTEGER 1024
2144 02 129: INTEGER
:          00 92 86 DB DA 91 EC CF C3 06 0A A5 59 83 18 E2
:          A6 39 F5 BA 90 A4 CA 65 61 57 B2 67 3F B1 91 CD
:          05 89 EE 05 F4 CE F1 BD 13 50 84 08 27 14 58 C3
:          08 51 CE 7A 4E F5 34 74 2B FB 11 F4 74 3C 8F 78
:          7B 11 19 3B A3 04 C0 E6 BC A2 57 01 BF 88 AF 1C
:          B9 B8 FD 47 11 D8 9F 88 E3 2B 37 D9 53 16 54 1B
:          F1 E5 DB B4 98 9B 3D F1 36 59 B8 8C 0F 97 A3 C1
:          08 7B 9F 2D 53 17 D5 57 DC D4 AF C6 D0 A7 54 E2

```

```

:       79
2276 02 33:  INTEGER
:       00 C9 66 E9 B3 B8 B7 CD D8 2F F0 F8 3A F8 70 36
:       C3 8F 42 23 8E C5 0A 87 6C D3 90 E4 3D 67 B6 01
:       3F
2311 02 128: INTEGER
:       7E 9C 30 96 67 6F 51 E3 B2 F9 88 4C F0 AC 21 56
:       77 94 96 F4 10 E0 49 CE D7 E5 3D 8B 7B 5B 36 6B
:       1A 60 08 E5 19 66 05 A5 5E 89 C3 19 0D AB F8 0B
:       9F 11 63 C9 79 FC D1 83 28 DA E5 E9 04 88 11 B3
:       70 10 7B B7 71 5F 82 09 1B B9 DE 0E 33 EE 2F ED
:       62 55 47 4F 87 69 FC E5 EA FA EE F1 CB 5A 32 E0
:       D5 C6 C2 F0 FC 0B 34 47 07 29 47 F5 B4 C3 87 66
:       69 93 A3 33 FC 06 56 8E 53 4A D5 6D 23 38 D7 29
2442 30 58:  SEQUENCE {
2444 06 7:   OBJECT IDENTIFIER id-GostR3410-94-bBis
2453 30 47:  SEQUENCE {
2455 02 4:   INTEGER 2046851076
2461 02 5:   INTEGER
:         00 D3 1A 4F F7
2468 02 32:  INTEGER
:         7E C1 23 D1 61 47 77 62 83 8C 2B EA 9D BD F3 30
:         74 AF 6D 41 D1 08 A0 66 A1 E7 A0 7A B3 04 8D E2
:         }
:       }
:     }
:   }
2502 30 380: SEQUENCE {
2506 06 7:   OBJECT IDENTIFIER
:         id-GostR3410-94-CryptoPro-XchC-ParamSet
2515 30 367: SEQUENCE {
2519 02 2:   INTEGER 1024
2523 02 129:  INTEGER
:         00 B1 94 03 6A CE 14 13 9D 36 D6 42 95 AE 6C 50
:         FC 4B 7D 65 D8 B3 40 71 13 66 CA 93 F3 83 65 39
:         08 EE 63 7B E4 28 05 1D 86 61 26 70 AD 7B 40 2C
:         09 B8 20 FA 77 D9 DA 29 C8 11 1A 84 96 DA 6C 26
:         1A 53 ED 25 2E 4D 8A 69 A2 03 76 E6 AD DB 3B DC
:         D3 31 74 9A 49 1A 18 4B 8F DA 6D 84 C3 1C F0 5F
:         91 19 B5 ED 35 24 6E A4 56 2D 85 92 8B A1 13 6A
:         8D 0E 5A 7E 5C 76 4B A8 90 20 29 A1 33 6C 63 1A
:         1D
2655 02 33:  INTEGER
:         00 96 12 04 77 DF 0F 38 96 62 8E 6F 4A 88 D8 3C
:         93 20 4C 21 0F F2 62 BC CB 7D AE 45 03 55 12 52
:         59
2690 02 128:  INTEGER
:         3F 18 17 05 2B AA 75 98 FE 3E 4F 4F C5 C5 F6 16

```



```

      :      E1 22 CF F9 EB D8 9E F8 1D C7 CE 8B F5 6C C6 4B
      :      43 58 6C 80 F1 C4 F5 6D D5 71 8F DD 76 30 0B E3
      :      36 78 42 59 CA 25 AA DE 5A 48 3F 64 C0 2A 20 CF
      :      4A 10 F9 C1 89 C4 33 DE FE 31 D2 63 E6 C9 76 46
      :      60 A7 31 EC CA EC B7 4C 82 79 30 37 31 E8 CF 69
      :      20 5B C7 3E 5A 70 BD F9 3E 5B B6 81 DA B4 EE B9
      :      C7 33 CA AB 2F 67 3C 47 5E 0E CA 92 1D 29 78 2E
2821 30 63: SEQUENCE {
2823 06 7:  OBJECT IDENTIFIER id-GostR3410-94-bBis
2832 30 52:  SEQUENCE {
2834 02 4:    INTEGER 371898640
2840 02 5:    INTEGER
      :      00 93 F8 28 D3
2847 02 37:  INTEGER
      :      00 CA 82 CC E7 8A 73 8B C4 6F 10 3D 53 B9 BF 80
      :      97 45 EC 84 5E 4F 6D A4 62 60 6C 51 F6 0E CF 30
      :      2E 31 20 4B 81
      :      }
      :    }
      :  }
      : }
      : }
      : }

```

```

|>GostR3410-94-ParamSetParameters.bin
|MIILQjCB0QYHKoUDAgIgADCBxQICAgACQQDugXKuiZZgj7aTWbieuCpPhUUQ4pd6
|TWO8lzIs5dwzhuoKERND6RkPIxd1OYRYOXhrsMNF0WWXbvIZXsmxw3njAiEAmJFe
|fsgmXt/Nox6I8kgJ3bBkvccoXdUNconwrG9J3S0CQQCelgMVAMh3SoaVgtSv3iEn
|r60l0LS2JwvpvIq3tQ1Q8gZlWYSknLCTBNZiViq1qrG0vizUasPYSVsUKqbOI+Ic
|MBYGBYqFAwICFAEwCwICXskCANNBAGECMIIBVgYHKoUDAgIgAjCCAUKCAGQAAoGB
|ALtiXvsBjJyLh1BeKmdVFP7cVsKRS35PidI/A/Azd+cKKQNIIndYOeEGNPYUe2lMX
|xIceQLBCKMO3kCljxLfYXVK5qojyr9vrKNqIadbFhGodmJJOk1VhvWkwC53dBdJH
|tZItlNy7AmcYgcV9EOXvctPm2tQiPcgqoffQKUZRpIDfAiEAlyQypDcXizC9lhlb
|dzeJqy//FVlLF23RdbYyVu5a8s8CgYEAj9NnMSN2VLvkH18fhFpNkHQU/8IsJdkV
|MJ5dLmKiomxxEfP8eVaNr6AoBC/hpSoEiYBcDemhpGnIRmfKu+5iXDB4iIwdhe6o
|g/GtW8Tmd26OGgdQkS32T3mVZJnx4YJHWwtg4mMq3NjPlOnFT9HzsQnYHwC/KrjL
|hirffUC5NpowGAYHKoUDAgIUBDANAgRSCHT1AgUA7jmtszCCAasGByqFAwICIAMw
|ggGeAgIEAAKBgQDGLx/FdSSzDJAYxeYh3hVJlzaFTlam+K7mWnpARjKxvPA0n/yv
|ywoQMxEXH8FhKtzbjIzJOMcCJcj9Eq/wGx0GTgrW/earkVkWbLny/BcdkvDMe2pr
|LNf6NcrL4skxWkLVdrHsznepYxV/PQvZao6wsPNQKtI4EBsFEWM08eW3qwIhALCd
|Y0wQiZzXlM0nZXQD4FgQsHxhpoi6ssN/Rl4wiwYHAoGAPSa0Z9lKP/ydcb+NuJNA
|hBNyZPPC6esW3KIUuLx8hySFM2dEk0/S711D+e0LdFuQqj7I1wzckWgkeLZkouH4
|+1b08pcv7n7bCEr3RkGbhU+tAsw+Nkb/LhoY3UvrPET38nRviAKWSWdFRsyRh8IH
|+48s70jikT9oOVxHBK8EurUwbgyYHKoUDAgIUBDBjAgRbl3zbAgRulpLdAlUAvDy7
|235vhIKG4ZrZonqOKX5bccU92XTN9g+TclbfacvJejAMzHFoXFUwRhR/EVaMT93z
|Y9nYhkODRaYsO3WWPWVGrfq/MbMSkNESrmXsUDCe9meCMIIBXwYHKoUDAgIgBDCC
|AVICAgQAAoGBAJ2I5tf+MxO9LnRcfN0que5K88iJnoR950ozeD6mi8MFiLofc4xq
|r4qzUFMfGFTDg3zDyGD/1+LhBsP2Oz2KTANM5zlCpsPVhbWZz2le16PEqTsrLhtx
|V7saHAQ6tB7IVmxhRek4phGQbedTLlYklFadfpmaDdpch5vdkf4STfHpAiEA+t0Z

```

```

|er0ZobRlPuz37KTWoisff4k7ZB+QFkH7tVU1T68CgYB0R+1xVjEFmQcLEmCZR6XI
|yKHiXPHPJStAezMfk9Y53dG6OSZW3sqZLdAlNUMpoelabjLW9HiC2WC48QrK/3lt
|E82WEfhT2rbSYjSD5GeIcISTk3oaKUQlMk7C4HQgIly0QP6cGHQOzmdlrAX68CSm
|SwJufkCIQIGeli5+X0Aa4zAiBgcqhQMCAhQEEMbcCBE0Eh0QCBQC1CoJtAgh/V16B
|lLxb3zCCAXMGBYqFAwICIAUwggFmAgIEAAKBgQCA8QLTKw/RZ9BpwnowetrSxGYJ
|GQTbqlXVuMxwJvL3ozGbiQy2UsQOBU4ekwZzW0PXSnnnt35ECABzZ4agx/ooWPu2J
|qwfPKr6CQqyd7d2/mNYS3dHqT18V06QqZne90pOyQmDA8nwPHRWUhhTVZ7ZvqQK6
|oRpprjvOrbuD45nJtQIhAPD1RMQYqsI09oPwM1EbZcIWUaYHi9otabufcyhnUCFJ
|AoGAa8wLT62ziJweBq3SPMCbirbs3t9z8EYyWV7kJQAF1q9fWt5Eyx4m5iY8ZyNH
|z6JvnpOTaB5rdZczeEzeXb2aFKOTad/Zn6hcwNECQcQBA0PzSpE5OnBs8SZ3y/of
|V41rbPvoosQs/MlLO2U6R24UXjhiwYzD/tglfP73TNSgW/EwNgYHKoUDAgIUBDar
|AgQT2oudAgUAoOneSwIcQauXhX9CYUNV0y2wsQafEJpNooNnbHxTpoGFtDCCAYwG
|ByqFAwICIQEwggF/AgIEAAKBgQDKoz8u7p/UYxfUlZWP51GObGPY9OtNItENKk8L
|iDnwefgonmA7AlMhHlM7Wh52hZ5IUMZwx7ccDfhMo+DWwXf+n3ip2EMyMKiDzYKi
|srXHozBpgCeFcM23m/AQdKacliM0iCSwxTerlTxqeMq2nhz7KDaGEaOX9Q9UHhbb
|NI2+XwIhAMrk2F+AwUdswykJX7AKkFeqSsxEzO4X8ZlctcVJpDZAoGBAL4nllLy
|8eM52nNCEbhbBq5N4jaqj77rPxrCxSzUOFN3foNkaLgBOGeKitvTpVxwp+qxunoH
|GVSGd6r05gn/tH9rnX5FsNBtg9etxTMQq9hXg+cx37HMmi2qcCNJguF2EhWlso5
|wXsX8ETR4FBikDar04HF5r+CujuGv8TZgVME4GBYqFAwICFAQwQwIFANBenxQC
|BEYwTF8CNDWrh1OZzaM8FGymKWYOWl5cb3FMoybbAy3WdRmVzBkKYSuSKJMtGwJW
|TsJKXe93OcWBPYmWggF3BgcqhQMCAiECMIIBagICBAACgYEAkobb2pHsz8MGCqVZ
|gxjipjnlupCkymVhV7JnP7GRzQWJ7gX0zvG9E1CECCcUWMMIUc56TvU0dCv7Efr0
|PI94exEZO6MEwOa8olcBv4ivHlM4/UcR2J+I4ys32VMWVBvx5du0mJs98TZZuIwP
|l6PBCHufLVMX1Vfc1K/G0KdU4nkCIQDJZumzuLfn2C/w+Dr4cDbDj0IjjsUKh2zT
|kOQ9Z7YBPwKBgH6cMjZnblHjsvmITPCsIVZ3lJb0EOBJztf1PYt7WzZrGmA15Rlm
|BaVeicMZDav4C58RY815/NGDKNr16QSIEbNwEHu3cV+CCRu53g4z7i/tYlVHT4dp
|/OXq+u7xyloy4NXGwvD8CzRHBylH9bTDh2Zpk6Mz/AZwjlnk1W0jONcpMDoGBYqF
|AwICFAQwLwIEegB4BAIFANMaT/cCIH7BI9FhR3dig4wr6p298zB0r2lB0QigZqHn
|ohqzBI3iMIIBfAYHKoUDAgIhAzCCAW8CAgQAAoGBALGUA2rOFBodNtZCla5sUPxL
|fWXYs0BxE2bKk/ODZTKI7mN75CgFHYZhJncte0AscBgg+nfz2inIERqEltpsJhpT
|7SUuTYppogN25q3b09zTMXSaSRoYS4/abYTDHPBfkRml7TukbqRWLYWYSi6ETao0O
|Wn5cdkuokCApoTNsYxodAIEAlhIEd98POJZijm9KiNg8kyBmiQ/yYrzLfa5FA1US
|UlkcGyA/GbCFK6p1mP4+T0/FxfYW4SLP+evYnvGdx86L9WzGS0NYbIDxxPVt1XGP
|3XYwC+M2eEJZyiWq3lpIP2TAKiDPSHD5wYnEM97+MdJj5s12RmCnMezK7LdMgnkw
|NzHoz2kgW8c+WnC9+T5btoHat065xzPKqy9nPEdeDsqsHS14LjA/BgcqhQMCAhQE
|MDQCBYquRACBQCT+CjTAiUAYoLM54pzi8RvED1Tub+Al0XshF5PbaRiYGxR9g7P
|MC4xIEuB
|<GostR3410-94-ParamSetParameters.bin

```

11.4. GOST R 34.10-2001 Public Key Algorithm Parameters

For each AlgorithmIdentifier in this sequence, the parameters field contains GostR3410-2001-ParamSetParameters.

```

0 30 998: SEQUENCE {
4 30 156: SEQUENCE {
7 06 7: OBJECT IDENTIFIER
      : id-GostR3410-2001-TestParamSet

```

```

16 30 144: SEQUENCE {
19 02 1:   INTEGER 7
22 02 32:   INTEGER
:         5F BF F4 98 AA 93 8C E7 39 B8 E0 22 FB AF EF 40
:         56 3F 6E 6A 34 72 FC 2A 51 4C 0C E9 DA E2 3B 7E
56 02 33:   INTEGER
:         00 80 00 00 00 00 00 00 00 00 00 00 00 00 00
:         00 00 00 00 00 00 00 00 00 00 00 00 00 00 04
:         31
91 02 33:   INTEGER
:         00 80 00 00 00 00 00 00 00 00 00 00 00 00 00
:         01 50 FE 8A 18 92 97 61 54 C5 9C FC 19 3A CC F5
:         B3
126 02 1:   INTEGER 2
129 02 32:   INTEGER
:         08 E2 A8 A0 E6 51 47 D4 BD 63 16 03 0E 16 D1 9C
:         85 C9 7F 0A 9C A2 67 12 2B 96 AB BC EA 7E 8F C8
:         }
:       }
163 30 159: SEQUENCE {
166 06 7:   OBJECT IDENTIFIER
:         id-GostR3410-2001-CryptoPro-A-ParamSet
175 30 147: SEQUENCE {
178 02 33:   INTEGER
:         00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF
:         FF FF FF FF FF FF FF FF FF FF FF FF FF FF FD
:         94
213 02 2:   INTEGER 166
217 02 33:   INTEGER
:         00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF
:         FF FF FF FF FF FF FF FF FF FF FF FF FF FF FD
:         97
252 02 33:   INTEGER
:         00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF
:         FF 6C 61 10 70 99 5A D1 00 45 84 1B 09 B7 61 B8
:         93
287 02 1:   INTEGER 1
290 02 33:   INTEGER
:         00 8D 91 E4 71 E0 98 9C DA 27 DF 50 5A 45 3F 2B
:         76 35 29 4F 2D DF 23 E3 B1 22 AC C9 9C 9E 9F 1E
:         14
:       }
:     }
325 30 188: SEQUENCE {
328 06 7:   OBJECT IDENTIFIER
:         id-GostR3410-2001-CryptoPro-B-ParamSet
337 30 176: SEQUENCE {
340 02 33:   INTEGER

```

```

:      00 80 00 00 00 00 00 00 00 00 00 00 00 00 00
:      00 00 00 00 00 00 00 00 00 00 00 00 00 00 0C
:      96
375 02  32:  INTEGER
:      3E 1A F4 19 A2 69 A5 F8 66 A7 D3 C2 5C 3D F8 0A
:      E9 79 25 93 73 FF 2B 18 2F 49 D4 CE 7E 1B BC 8B
409 02  33:  INTEGER
:      00 80 00 00 00 00 00 00 00 00 00 00 00 00 00
:      00 00 00 00 00 00 00 00 00 00 00 00 00 00 0C
:      99
444 02  33:  INTEGER
:      00 80 00 00 00 00 00 00 00 00 00 00 00 00 00
:      01 5F 70 0C FF F1 A6 24 E5 E4 97 16 1B CC 8A 19
:      8F
479 02   1:  INTEGER 1
482 02  32:  INTEGER
:      3F A8 12 43 59 F9 66 80 B8 3D 1C 3E B2 C0 70 E5
:      C5 45 C9 85 8D 03 EC FB 74 4B F8 D7 17 71 7E FC
:      }
:      }
516 30 159:  SEQUENCE {
519 06   7:  OBJECT IDENTIFIER
:      id-GostR3410-2001-CryptoPro-C-ParamSet
528 30 147:  SEQUENCE {
531 02  33:  INTEGER
:      00 9B 9F 60 5F 5A 85 81 07 AB 1E C8 5E 6B 41 C8
:      AA CF 84 6E 86 78 90 51 D3 79 98 F7 B9 02 2D 75
:      98
566 02   3:  INTEGER 32858
571 02  33:  INTEGER
:      00 9B 9F 60 5F 5A 85 81 07 AB 1E C8 5E 6B 41 C8
:      AA CF 84 6E 86 78 90 51 D3 79 98 F7 B9 02 2D 75
:      9B
606 02  33:  INTEGER
:      00 9B 9F 60 5F 5A 85 81 07 AB 1E C8 5E 6B 41 C8
:      AA 58 2C A3 51 1E DD FB 74 F0 2F 3A 65 98 98 0B
:      B9
641 02   1:  INTEGER 0
644 02  32:  INTEGER
:      41 EC E5 57 43 71 1A 8C 3C BF 37 83 CD 08 C0 EE
:      4D 4D C4 40 D4 64 1A 8F 36 6E 55 0D FD B3 BB 67
:      }
:      }
678 30 159:  SEQUENCE {
681 06   7:  OBJECT IDENTIFIER
:      id-GostR3410-2001-CryptoPro-XchA-ParamSet
690 30 147:  SEQUENCE {
693 02  33:  INTEGER

```

```

      :      00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
      :      FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FD
      :      94
728 02   2:  INTEGER 166
732 02   33: INTEGER
      :      00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
      :      FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FD
      :      97
767 02   33: INTEGER
      :      00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
      :      FF 6C 61 10 70 99 5A D1 00 45 84 1B 09 B7 61 B8
      :      93
802 02   1:  INTEGER 1
805 02   33: INTEGER
      :      00 8D 91 E4 71 E0 98 9C DA 27 DF 50 5A 45 3F 2B
      :      76 35 29 4F 2D DF 23 E3 B1 22 AC C9 9C 9E 9F 1E
      :      14
      :      }
      :      }
840 30  159: SEQUENCE {
843 06   7:  OBJECT IDENTIFIER
      :      id-GostR3410-2001-CryptoPro-XchB-ParamSet
852 30  147: SEQUENCE {
855 02   33: INTEGER
      :      00 9B 9F 60 5F 5A 85 81 07 AB 1E C8 5E 6B 41 C8
      :      AA CF 84 6E 86 78 90 51 D3 79 98 F7 B9 02 2D 75
      :      98
890 02   3:  INTEGER 32858
895 02   33: INTEGER
      :      00 9B 9F 60 5F 5A 85 81 07 AB 1E C8 5E 6B 41 C8
      :      AA CF 84 6E 86 78 90 51 D3 79 98 F7 B9 02 2D 75
      :      9B
930 02   33: INTEGER
      :      00 9B 9F 60 5F 5A 85 81 07 AB 1E C8 5E 6B 41 C8
      :      AA 58 2C A3 51 1E DD FB 74 F0 2F 3A 65 98 98 0B
      :      B9
965 02   1:  INTEGER 0
968 02   32: INTEGER
      :      41 EC E5 57 43 71 1A 8C 3C BF 37 83 CD 08 C0 EE
      :      4D 4D C4 40 D4 64 1A 8F 36 6E 55 0D FD B3 BB 67
      :      }
      :      }
      :      }

```

```

|>GostR3410-2001-ParamSetParameters.bin
|MIID5jCBnAYHKoUDAgIjADCBkAIBBwIgx7/0mKqTjOc5uOAi+6/vQFY/bmo0cvwq
|UUwM6driO34CIQCAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEMQIhAIAA
|AAAAAAAAAAAAAAAAAAAFQ/ooYkpdhVMWc/Bk6zPWzAgECAiAI4qig5lFHlL1jFgMO

```

```

|FtGchcl/CpyiZxIrlqu86n6PyDCBnwYHKoUDAgIjATCBkwIhAP////////////////
|////////////////2UAgIapgIhAP////////////////
|////////////////2XAiEA////////////////2xhEHCZWtEARYQbCbDhuJMC
|AQECIQCNkeRx4Jic2iffUFpFPyt2NSlPLd8j47EirMmncp8eFDCBvAYHkoUDAgIj
|AjCBsAIhAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAaYWAiA+GvQZomml
|+Gan08JcPfgK6Xklk3P/KxgvSdTOfhU8iwIhAIAAAAAAAAAAAAAAAAAAAAAAAAAA
|AAAAAAAAAAAAAAAAAaYZAiEAgAAAAAAAAAAAAAAAAAAAAAAAAAAV9wDP/xpiTl5JcWG8yKGY8C
|AQECID+oEkNZ+WAuD0cPrLAcOXFRcmFjQPs+3RL+NcXcX78MIGfBgcqhQMCAiMD
|MIGTAiEAm59gXlqFgQerHshea0HIqs+EboZ4kFHTeZj3uQItDzGCAwCAWgIhAJuf
|YF9ahYEHqx7IXmtByKrPhG6GeJBR03mY97kCLXWbAiEAm59gXlqFgQerHshea0HI
|qlgsolEe3ft08C86ZZiYC7kCAQACIEHs5VdDcRqMPL83g80IwO5NTcRA1GQajZu
|VQ39s7tnMIGfBgcqhQMCAiQAMIGTAiEA////////////////////////
|////////ZQCAgCmAiEA////////////////////////ZcC
|IQD////////////////bGEQcJla0QBfHbsJt2G4kwIBAqIhAI2R5HHgmJza
|J99QWku/K3Y1KU8t3yPjsSKsyZyenx4UMIGfBgcqhQMCAiQBMIGTAiEAm59gXlqF
|gQerHshea0HIqs+EboZ4kFHTeZj3uQItDzGCAwCAWgIhAJufYF9ahYEHqx7IXmtB
|yKrPhG6GeJBR03mY97kCLXWbAiEAm59gXlqFgQerHshea0HIqlgsolEe3ft08C86
|ZZiYC7kCAQACIEHs5VdDcRqMPL83g80IwO5NTcRA1GQajZuVQ39s7tn
|<GostR3410-2001-ParamSetParameters.bin

```

12. 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 aim 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.

Peter Gutmann for his helpful "dumpasn1" program.

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

Derek Atkins (IHTFP Consulting, derek@ihtfp.com) and his wife, Heather Anne Harrison, for making the document readable.

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 acknowledge CRYPTO-PRO. CRYPTO-PRO requests that all material mentioning or referencing this document identify this as "CRYPTO-PRO CPALGS".

13. References

13.1. 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)
- [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 34.11-94, Gosudarstvennyi Standard of Russian Federation, Government Committee of the Russia for Standards, 1994. (In Russian)
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [HMAC] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

13.2. Informative References

- [Schneier95] B. Schneier, Applied cryptography, second edition, John Wiley & Sons, Inc., 1995.
- [RFDSL] "Russian Federal Digital Signature Law", 10 Jan 2002 N 1-FZ
- [RFLLIC] "Russian Federal Law on Licensing of Selected Activity Categories", 08 Aug 2001 N 128-FZ

- [CRYPTOLIC] "Russian Federal Government Regulation on Licensing of Selected Activity Categories in Cryptography Area", 23 Sep 2002 N 691
- [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.
- [RFC4134] Hoffman, P., "Examples of S/MIME Messages", RFC 4134, July 2005.
- [TLS] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

Authors' Addresses

Vladimir Popov
CRYPTO-PRO
38, Obraztsova,
Moscow, 127018, Russian Federation

E-Mail: vpopov@cryptopro.ru

Igor Kurepkin
CRYPTO-PRO
38, Obraztsova,
Moscow, 127018, Russian Federation

E-Mail: kure@cryptopro.ru

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

E-Mail: lse@cryptopro.ru

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

E-Mail: chudov@cryptopro.ru

Alexandr Afanasiev
Factor-TS
office 711, 14, Presnenskij val,
Moscow, 123557, Russian Federation

E-Mail: afal@factor-ts.ru

Nikolaj Nikishin
Infotecs GmbH
p/b 35, 80-5, Leningradskij prospekt,
Moscow, 125315, Russian Federation

E-Mail: nikishin@infotecs.ru

Boleslav Izotov
FGUE STC "Atlas"
38, Obraztsova,
Moscow, 127018, Russian Federation

E-Mail: izotov@nii.voskhod.ru

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

E-Mail: evminaeva@mail.ru

Serguei Murugov
R-Alpha
4/1, Raspletina,
Moscow, 123060, Russian Federation

E-Mail: msm@top-cross.ru

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

E-Mail: igori@mo.msk.ru

Igor Ustinov
Cryptocom
office 239, 51, Leninskij prospekt,
Moscow, 119991, Russian Federation

E-Mail: igus@cryptocom.ru

Anatolij Erkin
SPRCIS (SPbRCZI)
1, Obrucheva,
St.Petersburg, 195220, Russian Federation

E-Mail: 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).

