Internet Draft
Vladimir Popov, CRYPTO-PRO
Igor Kurepkin, CRYPTO-PRO
Expires January 15, 2006
Serguei Leontiev, CRYPTO-PRO
Intended Category: Informational
July 15, 2005

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.

<draft-popov-cryptopro-cpalgs-03.txt>

Status of this Memo

Copyright Notice

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

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 need supplemental specifications 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 CRYPT-PRO Company to provide supplemental information and specifications needed by the "Russian Cryptographic Software Compatibility Agreement" community.

1.2 Terminology

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

in [RFC 2119].

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

'|' stands for concatenation.

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 as initialization vector.

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

gostR3411 (D) - is the 256-bit result of GOST R 34.11-94 hash function, used with zero intitialization vector, and S-Box parameter, defined by gostR3411CryptoProParamSetAI (see Appendix, GostR3411-94-ParamSetSyntax module).

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

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 key meshing algorithm. Key meshing transforms a symmetric key after some amount of data has been

processed. This document defines 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 to 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 to 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 < 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 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.

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 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 ::=
    { id-CryptoPro-algorithms 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, when some 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 such protocols as IPsec ESP requires special care.

The identifier for this algorithm is:

```
id-Gost28147-89-CryptoPro-KeyMeshing  OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms keyMeshing(14) cryptoPro(1) }
```

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

Encryption or decryption starts with key K[0] = K, IV0[0] = IV, i = 0. Let IV[0] be the value of the initialization vector after processing the first 1024 octets of data. Encryption or decryption of the next 1024 octets will start with K[1] and IV0[1], which are calculated using the formula:

```
K[i+1] = decryptECB (K[i], C);
IV0[i+1] = encryptECB (K[i+1],IV[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 every 1024 octets of data:

* the resulting initialization vector is stored as IV[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 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]. PRF_GOSTR3411(secret,label,seed) = P_GOSTR3411 (secret,label|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) (mod 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) = gostR3411 (K(x,y))

Keypairs x and y MUST comply with [GOSTR341094].

This algorithm MUST NOT be used when a^x = a (mod p) or a^y = a (mod 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)(mod q)) . (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 - User Keying Material (64 bit)
   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) = gostR3411 (K(x,y,UKM))

Keypairs x and y 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 identifier for this algorithm is:


```
id-Gost28147-89-None-KeyWrap OBJECT IDENTIFIER ::=
```

```
        { id-CryptoPro-algorithms keyWrap(13) none(0) }
```

The GOST 28147-89 key wrap algorithm is:

1) For a unique symmetric KEK, 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) 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 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 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 unique UKM is used to diversify the KEK.

Identifier for this algorithm:

```
    id-Gost28147-89-CryptoPro-KeyWrap OBJECT IDENTIFIER ::=
        { id-CryptoPro-algorithms keyWrap(13) cryptoPro(1) }
```

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 Unrap

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 error.
2) Decompose the 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 not equal, then 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]|..|a[7] (a[i] - byte, a[i,0]..a[i,7] - it's bits)

3) Let i be 0.

4) K[1]..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]|..|k[i,7] (k[i,j] - 32-bit integer)
B) Vector S[i] is calculated:
S[i] = ((a[i,0]*k[i,0] + ... + a[i,7]*k[i,7]) mod 2^32)
| ((~a[i,0]*k[i,0] + ... + ~a[i,7]*k[i,7]) mod 2^32);
C) K[i+1] = 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 Kd, 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 4-bytes long, B = D|D|D|D|D|D|D|D|D|D.

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 XOR UKM[i]) is used.
   * During each encryption step, only 8 out of 32 GOST 28147-89 rounds are done.

4) Kd is calculated:
   Kd = 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 and corresponding parameter values for all of the proposed parameter sets can be found in the Appendix in the form of ASN.1 modules [X.660].

8.1 Encryption algorithm parameters

GOST 28147-89 can be used in several modes, additional CBC mode is defined in section 2.1 this document. It also has an S-Box parameter (see Algorithm Parameters part in [GOST28147] in Russian, description

in English see in [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  } |
     { Gost28147-89-ParamSetParameters IDENTIFIED BY
            id-Gost28147-89-CryptoPro-Simple-A-ParamSet  } |
     { Gost28147-89-ParamSetParameters IDENTIFIED BY
            id-Gost28147-89-CryptoPro-Simple-B-ParamSet  } |
     { Gost28147-89-ParamSetParameters IDENTIFIED BY
            id-Gost28147-89-CryptoPro-Simple-C-ParamSet  } |
     { Gost28147-89-ParamSetParameters IDENTIFIED BY
            id-Gost28147-89-CryptoPro-Simple-D-ParamSet  }
 }
```

Identifier values are in the Appendix.

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) },
     keyWrap       AlgorithmIdentifier,
     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 }
 }
 Gost28147-89-KeyWrapAlgorithms  ALGORITHM-IDENTIFIER ::= {
     { NULL IDENTIFIED BY id-Gost28147-89-CryptoPro-KeyWrap } |
     { NULL IDENTIFIED BY id-Gost28147-89-None-KeyWrap }
 }
```

where
   eUZ   - S-box value;
   mode   - cipher mode;
   shiftBits - cipher parameter;
   keyWrap   - key export algorithm identifier;
   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.

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.

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 (mod 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.

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

Cryptographic algorithm parameters affect rigidity of algorithms. The algorithm parameters proposed and described herein have been analyzed by special certification laboratory of Scientific and Technical Center "ATLAS" and by Center of Certificational Investigations in appropriate levels of target_of_evaluation (TOE), according to [RFDSL], [RFLLIC] and [CRYPTOLIC].

Use of different parameter sets is NOT RECOMENDED. When different parameters are used it is RECCOMENDED to subject them 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 which 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-TransportLayerSecurity OBJECT IDENTIFIER ::=
        { id-CryptoPro-modules gost-CryptoPro-TransportLayerSecurit
y(16) 1 }

  -- Crypto-Pro Policy
    gost-CryptoPro-Policy OBJECT IDENTIFIER ::=
        { id-CryptoPro-modules gost-CryptoPro-Policy(17) 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 which 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-ParamSet
        OBJECT IDENTIFIER ::=
           { id-CryptoPro-encrypts cryptopro-Oscar(5) }
    id-Gost28147-89-CryptoPro-Simple-A-ParamSet
        OBJECT IDENTIFIER ::=
           { id-CryptoPro-encrypts cryptopro-Simple-A(6) }
    id-Gost28147-89-CryptoPro-Simple-B-ParamSet
        OBJECT IDENTIFIER ::=
           { id-CryptoPro-encrypts cryptopro-Simple-B(7) }
    id-Gost28147-89-CryptoPro-Simple-C-ParamSet
        OBJECT IDENTIFIER ::=
           { id-CryptoPro-encrypts cryptopro-Simple-C(8) }
    id-Gost28147-89-CryptoPro-Simple-D-ParamSet
        OBJECT IDENTIFIER ::=
           { id-CryptoPro-encrypts cryptopro-Simple-D(9) }
  -- GOST 28147-89 Types
    Gost28147-89-Data ::= OCTET STRING (SIZE (0..4294967294))
    Gost28147-89-EncryptedData ::=
```

```
        OCTET STRING (SIZE (0..4294967294))
    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 OPTI
ONAL,
            macKey               Gost28147-89-MAC (SIZE (4))
        }
    Gost28147-89-BlobParameters ::=
        SEQUENCE {
            encryptionParamSet
```



```
                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-Simple-A-ParamSet |
                    id-Gost28147-89-CryptoPro-Simple-B-ParamSet |
                    id-Gost28147-89-CryptoPro-Simple-C-ParamSet |
                    id-Gost28147-89-CryptoPro-Simple-D-ParamSet
                ),
            ...
        }
  -- GOST 28147-89 encryption algorithm parameters
    Gost28147-89-Parameters ::=
        SEQUENCE {
            iv                   Gost28147-89-IV,
            encryptionParamSet
                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-Simple-A-ParamSet |
                    id-Gost28147-89-CryptoPro-Simple-B-ParamSet |
                    id-Gost28147-89-CryptoPro-Simple-C-ParamSet |
                    id-Gost28147-89-CryptoPro-Simple-D-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 which 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,
        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,
        id-Gost28147-89-CryptoPro-Simple-A-ParamSet,
        id-Gost28147-89-CryptoPro-Simple-B-ParamSet,
        id-Gost28147-89-CryptoPro-Simple-C-ParamSet,
        id-Gost28147-89-CryptoPro-Simple-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) },
        keyWrap         AlgorithmIdentifier,
        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  } |
    { Gost28147-89-ParamSetParameters IDENTIFIED BY
                id-Gost28147-89-CryptoPro-Simple-A-ParamSet  } |
    { Gost28147-89-ParamSetParameters IDENTIFIED BY
```

```
                id-Gost28147-89-CryptoPro-Simple-B-ParamSet  } |
    { Gost28147-89-ParamSetParameters IDENTIFIED BY
                id-Gost28147-89-CryptoPro-Simple-C-ParamSet  } |
    { Gost28147-89-ParamSetParameters IDENTIFIED BY
                id-Gost28147-89-CryptoPro-Simple-D-ParamSet  }
  }
  id-Gost28147-89-CryptoPro-KeyWrap OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms keyWrap(13) cryptoPro(1) }
  id-Gost28147-89-None-KeyWrap OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms keyWrap(13) none(0) }
  Gost28147-89-KeyWrapAlgorithms  ALGORITHM-IDENTIFIER ::= {
    { NULL IDENTIFIED BY id-Gost28147-89-CryptoPro-KeyWrap } |
    { NULL IDENTIFIED BY id-Gost28147-89-None-KeyWrap }
  }
  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 }
  }
  -- GOST 28147-89 cryptographic parameter set: values
    -- Test parameter set
  gost28147-89-UZ-Test Gost28147-89-UZ ::=
   '4CDE389C2989EFB6FFEB56C55EC29B029875613B113F896003970C798AA1D55
DE210AD43375DB38EB42C77E7CD46CAFAD66A201F70F41EA4AB03F22165B844D8'H
  gost28147-89-TestParamSetAI
    AlgorithmIdentifier ::= {
      algorithm
        id-Gost28147-89-TestParamSet,
      parameters
        Gost28147-89-ParamSetParameters:{
          eUZ           gost28147-89-UZ-Test,
          mode          gost28147-89-CNT,
          shiftBits     64,
```



```
          keyWrap       { algorithm id-Gost28147-89-None-KeyWrap },
          keyMeshing    { algorithm id-Gost28147-89-None-KeyMeshing
 }
        }
    }
```

```
  -- CryptoPro parameter sets
  gost28147-89-UZ-CryptoPro-A Gost28147-89-UZ ::=
   -- 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
   '93EEB31B67475ADA3E6A1D2F292C9C9588BD8170BA31D2AC1FD3F06E70890B0
8A5C0E78642F245C2E65B2943FCA43459CB0FC8F104787F37DD15AEBD519666E4'H
    gost28147-89-CryptoPro-A-ParamSetAI
        AlgorithmIdentifier  ::=
        {
            algorithm
                id-Gost28147-89-CryptoPro-A-ParamSet,
            parameters
                Gost28147-89-ParamSetParameters:{
                    eUZ         gost28147-89-UZ-CryptoPro-A,
                    mode        gost28147-89-CFB,
                    shiftBits   64,
                    keyWrap
                        { algorithm id-Gost28147-89-CryptoPro-KeyWr
ap },
                    keyMeshing
                        { algorithm id-Gost28147-89-CryptoPro-KeyMe
shing }
                }
        }
    --
  gost28147-89-UZ-CryptoPro-B Gost28147-89-UZ ::=
   -- K1 K2 K3 K4 K5 K6 K7 K8
   -- 8  0  E  7  2  8  5  0
```

```
   -- 4  1  C  5  7  3  2  4
   -- B  2  0  0  C  2  A  B
   -- 1  A  A  D  F  6  B  E
   -- 3  4  9  B  9  4  9  8
   -- 5  D  2  6  5  D  1  3
   -- 0  5  D  1  A  E  C  7
   -- 9  C  B  2  B  B  3  1
   -- 2  9  7  3  1  C  7  A
   -- E  7  5  A  4  1  4  2
   -- A  3  8  C  0  7  D  9
   -- C  F  F  F  D  F  0  6
   -- D  B  3  4  6  A  6  F
   -- 6  8  6  E  8  0  F  D
   -- 7  6  1  9  E  9  8  5
   -- F  E  4  8  3  5  E  C
   '80E7285041C57324B200C2AB1AADF6BE349B94985D265D1305D1AEC79CB2BB3
129731C7AE75A4142A38C07D9CFFFDF06DB346A6F686E80FD7619E985FE4835EC'H
    gost28147-89-CryptoPro-B-ParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-Gost28147-89-CryptoPro-B-ParamSet,
            parameters
                Gost28147-89-ParamSetParameters:{
                    eUZ         gost28147-89-UZ-CryptoPro-B,
                    mode        gost28147-89-CFB,
                    shiftBits   64,
                    keyWrap
                        { algorithm id-Gost28147-89-CryptoPro-KeyWr
ap },
                    keyMeshing
                        { algorithm id-Gost28147-89-CryptoPro-KeyMe
shing }
                }
        }
    --
  gost28147-89-UZ-CryptoPro-C Gost28147-89-UZ ::=
   -- K1 K2 K3 K4 K5 K6 K7 K8
   -- 1  0  8  3  8  C  A  7
   -- B  1  2  6  D  9  9  4
   -- C  7  5  0  B  B  6  0
   -- 2  D  0  1  0  1  8  5
   -- 9  B  4  5  4  8  D  A
   -- D  4  9  D  5  E  E  2
   -- 0  5  F  A  1  2  2  F
   -- F  2  A  8  2  4  0  E
   -- 4  8  3  B  9  7  F  C
   -- 5  E  7  2  3  3  3  6
```

```
   -- 8  F  C  9  C  6  5  1
   -- E  C  D  7  E  5  B  B
   -- A  9  6  E  6  A  4  D
   -- 7  A  E  F  F  0  1  9
   -- 6  6  1  C  A  F  C  3
   -- 3  3  B  4  7  D  7  8
   '10838CA7B126D994C750BB602D0101859B4548DAD49D5EE205FA122FF2A8240
E483B97FC5E7233368FC9C651ECD7E5BBA96E6A4D7AEFF019661CAFC333B47D78'H
    gost28147-89-CryptoPro-C-ParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-Gost28147-89-CryptoPro-C-ParamSet,
            parameters
                Gost28147-89-ParamSetParameters:{
                    eUZ         gost28147-89-UZ-CryptoPro-C,
                    mode        gost28147-89-CFB,
                    shiftBits   64,
                    keyWrap
                        { algorithm id-Gost28147-89-CryptoPro-KeyWr
ap },
                    keyMeshing
                        { algorithm id-Gost28147-89-CryptoPro-KeyMe
shing }
                }
        }
    --
  gost28147-89-UZ-CryptoPro-D Gost28147-89-UZ ::=
   -- K1 K2 K3 K4 K5 K6 K7 K8
   -- F  B  1  1  0  8  3  1
   -- C  6  C  5  C  0  0  A
   -- 2  3  B  E  8  F  6  6
   -- A  4  0  C  9  3  F  8
   -- 6  C  F  A  D  2  1  F
   -- 4  F  E  7  2  5  E  B
   -- 5  E  6  0  A  E  9  0
   -- 0  2  5  D  B  B  2  4
   -- 7  7  A  6  7  1  D  C
   -- 9  D  D  2  3  A  8  3
   -- E  8  4  B  6  4  C  5
   -- D  0  8  4  5  7  4  9
   -- 1  5  9  9  4  C  B  7
```

```
   -- B  A  3  3  E  9  A  D
   -- 8  9  7  F  F  D  5  2
   -- 3  1  2  8  1  6  7  E'H
   'FB110831C6C5C00A23BE8F66A40C93F86CFAD21F4FE725EB5E60AE90025DBB2
477A671DC9DD23A83E84B64C5D084574915994CB7BA33E9AD897FFD523128167E'H
    gost28147-89-CryptoPro-D-ParamSetAI
```



```
        AlgorithmIdentifier ::=
        {
            algorithm
                id-Gost28147-89-CryptoPro-D-ParamSet,
            parameters
                Gost28147-89-ParamSetParameters:{
                    eUZ         gost28147-89-UZ-CryptoPro-D,
                    mode        gost28147-89-CFB,
                    shiftBits   64,
                    keyWrap
                        { algorithm id-Gost28147-89-CryptoPro-KeyWr
ap },
                    keyMeshing
                        { algorithm id-Gost28147-89-CryptoPro-KeyMe
shing }
                }
        }
    --
    gost28147-89-CryptoPro-Simple-A-ParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-Gost28147-89-CryptoPro-Simple-A-ParamSet,
            parameters
                Gost28147-89-ParamSetParameters:{
                    eUZ         gost28147-89-UZ-CryptoPro-A,
                    mode        gost28147-89-CFB,
                    shiftBits   64,
                    keyWrap
                        { algorithm id-Gost28147-89-None-KeyWrap },
                    keyMeshing
                        { algorithm id-Gost28147-89-CryptoPro-KeyMe
shing }
                }
        }
```

```
    --
    gost28147-89-CryptoPro-Simple-B-ParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-Gost28147-89-CryptoPro-Simple-B-ParamSet,
            parameters
                Gost28147-89-ParamSetParameters:{
                    eUZ         gost28147-89-UZ-CryptoPro-B,
                    mode        gost28147-89-CFB,
                    shiftBits   64,
                    keyWrap
                        { algorithm id-Gost28147-89-None-KeyWrap },
```



```
                    keyMeshing
                        { algorithm id-Gost28147-89-CryptoPro-KeyMe
shing }
                }
        }
    --
    gost28147-89-CryptoPro-Simple-C-ParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-Gost28147-89-CryptoPro-Simple-C-ParamSet,
            parameters
                Gost28147-89-ParamSetParameters:{
                    eUZ         gost28147-89-UZ-CryptoPro-C,
                    mode        gost28147-89-CFB,
                    shiftBits   64,
                    keyWrap
                        { algorithm id-Gost28147-89-None-KeyWrap },
                    keyMeshing
                        { algorithm id-Gost28147-89-CryptoPro-KeyMe
shing }
                }
        }
    --
    gost28147-89-CryptoPro-Simple-D-ParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
```

```
                id-Gost28147-89-CryptoPro-Simple-D-ParamSet,
            parameters
                Gost28147-89-ParamSetParameters:{
                    eUZ         gost28147-89-UZ-CryptoPro-D,
                    mode        gost28147-89-CFB,
                    shiftBits   64,
                    keyWrap
                        { algorithm id-Gost28147-89-None-KeyWrap },
                    keyMeshing
                        { algorithm id-Gost28147-89-CryptoPro-KeyMe
shing }
                }
        }
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 which 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,
        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-Data, Gost28147-89-UZ
        FROM Gost28147-89-EncryptionSyntax
```

```
            gost28147-89-EncryptionSyntax
    ;
  -- 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-Data ::= Gost28147-89-Data
    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 which 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,
        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-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
        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 34.11-94 cryptographic parameter sets:
  -- OIDs for parameter sets are imported from GostR3411-94-DigestS
yntax
    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
        }
    }
    -- GOST R 34.11-94 Tests parameter set
    -- (GOST R 34.11-94 Annex A. Test vector)
```

```
    gostR3411TestParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3411-94-TestParamSet,
            parameters
                GostR3411-94-ParamSetParameters:{
                    hUZ
                       -- 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
                         '4E5764D1AB8DCBBF941A7A4D2CD11010D6A057358
D38F2F70F49D15AEA2F8D9462EE4309B3F4A6A218C698E3C17CE57E706B0966F702
3C8B5595BF2839B32ECC'H,
                    h0   '0000000000000000000000000000000000000000000
0000000000000000000000'H
                }
        }
    -- CryptoPro parameters
    gostR3411CryptoProParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3411-94-CryptoProParamSet,
            parameters
                GostR3411-94-ParamSetParameters:{
                    hUZ
```

```
                        -- pi1 pi2 pi3 pi4 pi5 pi6 pi7 pi8
                        -- A   5   7   4   7   7   D   1
                        -- 4   F   F   A   6   6   E   3
                        -- 5   4   C   7   4   2   4   A
                        -- 6   0   E   C   B   4   1   9
                        -- 8   2   9   0   9   D   7   5
                        -- 1   D   4   F   C   9   0   B
                        -- 3   B   1   2   2   F   5   4
                        -- 7   9   0   8   A   0   A   F
                        -- D   1   3   E   1   A   3   8
                        -- C   7   B   1   8   1   C   6
                        -- E   6   5   6   0   5   8   7
                        -- 0   3   2   5   E   B   F   E
                        -- 9   C   6   D   F   8   6   D
                        -- 2   E   A   B   D   E   2   0
                        -- B   A   8   9   3   C   9   2
                        -- F   8   D   3   5   3   B   C
                          'A57477D14FFA66E354C7424A60ECB41982909D751
D4FC90B3B122F547908A0AFD13E1A38C7B181C6E65605870325EBFE9C6DF86D2EAB
DE20BA893C92F8D353BC'H,
                      h0   '0000000000000000000000000000000000000000
0000000000000000000000'H
                  }
        }
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 which 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 }
        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,
        id-Gost28147-89-CryptoPro-Simple-A-ParamSet,
        id-Gost28147-89-CryptoPro-Simple-B-ParamSet,
        id-Gost28147-89-CryptoPro-Simple-C-ParamSet,
        id-Gost28147-89-CryptoPro-Simple-D-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-PublicKeyOctetString ::=
```



```
        OCTET STRING ( SIZE(
                        64 |    -- Only for testing purposes
                        128
                        ) )
    GostR3410-94-PublicKey ::=
        BIT STRING ( SIZE(16..1048) )
            -- Container for GostR3410-94-PublicKeyOctetString
    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
                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-Simple-A-ParamSet |
```

```
                    id-Gost28147-89-CryptoPro-Simple-B-ParamSet |
                    id-Gost28147-89-CryptoPro-Simple-C-ParamSet |
                    id-Gost28147-89-CryptoPro-Simple-D-ParamSet
                ) OPTIONAL
        }
    GostR3410-94-PublicKeyAlgorithms  ALGORITHM-IDENTIFIER ::= {
        { GostR3410-94-PublicKeyParameters IDENTIFIED BY
                        id-GostR3410-94 }
    }
    GostR3410-94-CertificateSignatureAlgorithms
        ALGORITHM-IDENTIFIER ::= {
                { NULL IDENTIFIED BY
```



```
                        id-GostR3411-94-with-GostR3410-94 } |
                { GostR3410-94-PublicKeyParameters IDENTIFIED BY
                        id-GostR3411-94-with-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 which 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-PKISynt
```



```
ax
    GostR3410-94-ParamSetParameters ::=
        SEQUENCE {
            t    INTEGER (512 | 1024),
                -- 512 - only for testing purposes
            p    INTEGER (
                        167597599124282463744675312477573076593492072757404917221544518046522050375919337210023428727086292846125398227331075635671923535149332124330420612576051361343078079299425970995740249982058461274793658205923933777235614437217640300735469768018742981669034276900318581864860508537538828119465699464336490060840951123558209288947442330815744243140458511235611838941607958938007235829223784381019579427983265047100132000711749196208485367436055090103890580296441496713277361049333905409282976888872507788088246581768450531286055238441764640393009211956940880170232270940691778664363999670287115498226905220977060151400857797693134862315907729305190789024733617976978942306572734300811577326758055009631327084773224075360211201138
```

```
9871393357658789768814416622492847430639474124377767893424865485276
3022196012460941194530829520850057688381506823424628814739131105408
2723716335051068458629823994724593847971630483535632962422413721 5
                        ),
                -- 2^509 < p < 2^512 or 2^1020 < p < 2^1024
            q       INTEGER (
                        2894802230932904885589274625217197696331749
6166410141009864396001978282409985
                        ..
                        1157920892373161954235709850086879078532699
8466564056403945758400791312963993 5
                        ),
                -- 2^254 < q < 2^256
            a       INTEGER (
                        2
                        ..
                        1797693134862315907729305190789024733617976
9789423065727343008115773267580550096313270847732240753602112011387
9871393357658789768814416622492847430639474124377767893424865485276
3022196012460941194530829520850057688381506823424628814739131105408
2723716335051068458629823994724593847971630483535632962422413721 4
                        ),
                -- 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 ::=
        SEQUENCE {
            x0   INTEGER (0 .. 65535),
            c    INTEGER (0 .. 65535),
            d    INTEGER (
                        2
                        ..
                        1797693134862315907729305190789024733617976
978942306572734300811577326758055009631327084773224075360211201138 7
987139335765878976881441662249284743063947412437776789342486548527 6
302219601246094119453082952085005768838150682342462881473913110540 8
272371633505106845862982399472459384797163048353563296242241372 14
                        )       -- 1 < d < p-1 < 2^1024-1
                        OPTIONAL
        }
    GostR3410-94-ValidationBisParameters ::=
        SEQUENCE {
            x0   INTEGER (0 .. 4294967295),
            c    INTEGER (0 .. 4294967295),
            d    INTEGER (
```



```
                        2
                        ..
                        1797693134862315907729305190789024733617976
978942306572734300811577326758055009631327084773224075360211201138 7
987139335765878976881441662249284743063947412437776789342486548527 6
302219601246094119453082952085005768838150682342462881473913110540 8
272371633505106845862982399472459384797163048353563296242241372 14
                        )       -- 1 < d < p-1 < 2^1024-1
```

```
                        OPTIONAL
        }
    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 }
    }
  -- GOST R 34.10-94 keys parameter sets
    -- GOST R 34.10-94 Tests parameter set
    -- (GOST R 34.10-94 Annex A. Test vector)
    gostR3410-94-TestParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3410-94-TestParamSet,
            parameters
                GostR3410-94-ParamSetParameters:{
                    t       512,
                    p       12491554796616397392007291845361681019980789084728846304013646795466302633346425772369277064638881858428879662416202925770315709968465491470753112581700067,
                    q       690083979912374782185295287117535788574643562215565368387576361326463015887 81,
                    a       83058219567796281938527505088117572448899826328218435214910357131733714685287987538317442674072307045274610623217326690344327461737869581425729297724134 68,
                    validationAlgorithm {
                        algorithm
                            id-GostR3410-94-a,
                        parameters
                            GostR3410-94-ValidationParameters: {
                                x0      24265,
                                c       29505,
                                d       2
                            }
```



```
                    }
```

```
                }
        }
    -- CryptoPro parameters
    gostR3410-94-CryptoPro-A-ParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3410-94-CryptoPro-A-ParamSet,
            parameters
                GostR3410-94-ParamSetParameters:{
                    t       1024,
                    p       127021248288932417465907042777176443525
78765350891653581281750726570503126098509849742318833348340118092599999512098893413065920561499672425412104927434935707492031276956145168922411057931124881261022967853463840169352001328899500036226068422275081353230700451734163368500454106258697141688368677884253782038383,
                    q       68363196144955700784444165611827252895102170888761442055095051287550314083023,
                    a       10099790675505530477208181553592522486984108257205345787482351587557714799052927277724415285269929879648335669968284202797289605274717317548059048560713474685214192868091256150280222218564753919090265611636784727014501906679429093018544621639973087222173288983032319409735540321340097258832287685094674066396262,
                    validationAlgorithm {
                        algorithm
                            id-GostR3410-94-bBis,
                        parameters
                            GostR3410-94-ValidationBisParameters: {
                                x0      1376285941,
                                c       3996757427
                            }
                    }
                }
        }
    --
    gostR3410-94-CryptoPro-B-ParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3410-94-CryptoPro-B-ParamSet,
            parameters
                GostR3410-94-ParamSetParameters:{
                    t       1024,
                    p       139454871199115825601409655107690713107041707059928031797758001454375765357722984094124368522288239833039 1
```

```
14681648076688236921220737322672160740747771700911134550432053804647694904686120113087816240740184800477047157336662926249423571248823968542221753660143391485680840520336859458494803187341288580489525163,
                    q       7988514166341097689762711893575632374730795191650763975830047269233887353395 9,
                    a       4294182614861580414387344773795550239267234596860714306679811299408947123142002706038521669956384871995765728481489890977075946261343766945636488273037083893479108083593264797677860191534347440096103423131667257868692048219493287863336020338479709268434224762105576023501613261478065276102850944540333865234 1,
                    validationAlgorithm {
                        algorithm
                            id-GostR3410-94-bBis,
                        parameters
                            GostR3410-94-ValidationBisParameters: {
                                x0      1536654555,
                                c       1855361757,
                                d       1440862938614001456765549029392820565478578022414617829967020177130599747551043947399151406115284791024439062735788342744854120601660303926203867703556828005895720381811489539897659442553756127180085030 6
                                }
                        }
                    }
        }
    --
    gostR3410-94-CryptoPro-C-ParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3410-94-CryptoPro-C-ParamSet,
            parameters
                GostR3410-94-ParamSetParameters:{
                    t       1024,
                    p       1106246792335119630405189524170170402485862954819831383774196396298584395948970608956170224210628525560327863824671665543929765440292184474789307951866999282788079219299270114285465514338758063771104435342935540667126530349962770993207157743542287621283671843703709141350171945045805050291770503634517804938 01,
                    q       1134688611998193505648682333788751980432679477648851099796123167253289954910 3,
                    a       8165527179708810160178931914153003482262544051353358162468249467681876621283478212884286545844013955142622
```

```
208772348502372286802227500950222482786620174449402169771648200835
639820229802489262048089869933550806433231352972533220881945689510 8
```



```
515517810022100345937058829107307118655300596214993684073712871083
2,
                    validationAlgorithm {
                        algorithm
                            id-GostR3410-94-bBis,
                        parameters
                            GostR3410-94-ValidationBisParameters: {
                                x0      1132758852,
                                c       3037364845,
                                d       9175906676429839327
                            }
                    }
                }
        }
    --
    gostR3410-94-CryptoPro-D-ParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3410-94-CryptoPro-D-ParamSet,
            parameters
                GostR3410-94-ParamSetParameters:{
                    t       1024,
                    p       905457649621929965904290958774625315611
305608390738976697140481252442226251255605447462085599609157078671 3
584955023674191558418599062780106646580951009578471398981941382087 1
596464891449305340792073707889052048273062303883776771017366483823 9
857482878789128647120146047432661269784969366551807386443649789321 4
9,
                    q       108988435796353506912374591498972192620
190487557619582334771735390599299211593,
                    a       756976611021707301782128757801610628085
528380310957115882957428141920853258904166001701785985821634140037 1
468755141279440056287893526663075439267701459858210336598311917392 4
473251122546471225238680331590270772766871534347608635047202529828 2
727146169012505061685823838436633108977746354101303392672374325483 3
7,
                    validationAlgorithm {
                        algorithm
```

```
                        id-GostR3410-94-bBis,
                    parameters
                        GostR3410-94-ValidationBisParameters: {
                            x0      333089693,
                            c       2699681355,
                            d       691588776390130148119174466
52402788947864438221427558424603662432 52
                            }
                    }
```



```
                }
        }
    --
    gostR3410-94-CryptoPro-XchA-ParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3410-94-CryptoPro-XchA-ParamSet,
            parameters
                GostR3410-94-ParamSetParameters:{
                    t       1024,
                    p       142011741597563481196368286022318089743
276138395243738762872573441927459393512718973631166078467600360848 9
466235676257952827747192122419290710461342083806363940845126918288 9
400057152462544529576934935675272895683154177544176313938445719175 5
096847107846595662547942312293338483924514339614727760681880609734 2
39,
                    q       917715298965546059455881490183827502172
968583935207241727433257254743749798 01,
                    a       133531813272720673433859519948319001217
942375967847486899482359599369642528734712461590403327731821410328 0
125292538719147885989931033105677441361963648030647213778266568986 8
646846327771015080940118260877020161532499046833293129492091277624 1
137878030224355746606283971659376426832674269780880061631528163475 8
87,
                    validationAlgorithm {
                        algorithm
                            id-GostR3410-94-bBis,
                        parameters
                            GostR3410-94-ValidationBisParameters: {
                                x0      3495862036,
                                c       1177570399,
```

```
                                d       354788961024091889513964706
4772083281962391865341410582282334567466222018672580177997251216990
526446086243776416033483110745 9
                                }
                            }
                        }
                }
    --
    gostR3410-94-CryptoPro-XchB-ParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3410-94-CryptoPro-XchB-ParamSet,
            parameters
                GostR3410-94-ParamSetParameters:{
                    t       1024,
```



```
                    p       1028946126624994859676552074360530315217
970499989304888248413244847492302275847016799887100360467070487737
728617617122769409863315390895687841291101095126905033453938698712 9
578346725726486834172001966298605611936667524296823673970848151797 5
203642359573653368957392061769855284593965042530895046088067160269 4
33,
                    q       9109671391802626916582318050603555673622
876949818259308838879688852816415951 99,
                    a       8890864727828423151699995801875757891030
146333865257914005197365930481314406858570673698294079477444963066 5
629150550360825239944379002723867491459962308678322286619775439928 1
674525482329862985987535754662860517388378547361676857690177803358 0
451144077333719625384235329193944778736647528245099866178789924431 7,
                    validationAlgorithm {
                        algorithm
                            id-GostR3410-94-bBis,
                        parameters
                            GostR3410-94-ValidationBisParameters: {
                                x0      2046851076,
                                c       3541716983,
                                d       573326676109894760566159697
288915335660587873174927484418272365769042745461 46
                                }
                    }
```

```
                }
        }
    --
    gostR3410-94-CryptoPro-XchC-ParamSetAI
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3410-94-CryptoPro-XchC-ParamSet,
            parameters
                GostR3410-94-ParamSetParameters:{
                    t       1024,
                    p       124699636699347751360714726579406443620
340886139505598921724845572998707376989996514806623647239928593208 6
882284875116543835094332766472226259406155605804500409472118260277 2
997756354023716906304480797157716494477784470005974190324577222262 5
326969837444652835352729304393746106576383349151001715930924115499 5
49,
                    q       678787613733659123438029502006568252711
812946805014794311467542947484224927 61,
                    a       443061846429758418247313503080985932686
399065011894175699527007486099731814269502352396232391105574508269 1
929579287893875210186770471816232510275169531004318559648376026578 2
782819424960556189369658653255131371944831362477736534684101187967 4


Popov,Kurepkin,Leontiev       Informational                    [Page 41]
```

```
Internet-Draft     Crypto-Pro cryptographic algorithms         July 2005


070984082549699793755607223451067047210860259793099687631930729083 3
4,
                    validationAlgorithm {
                        algorithm
                            id-GostR3410-94-bBis,
                        parameters
                            GostR3410-94-ValidationBisParameters: {
                                x0      371898640,
                                c       2482514131,
                                d       393411701713094918946116909
229454740026575590650016887148241594213466186452691964676993
                                }
                    }
                }
        }
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 which 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 }
        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,
```



```
        id-Gost28147-89-CryptoPro-Simple-A-ParamSet,
        id-Gost28147-89-CryptoPro-Simple-B-ParamSet,
        id-Gost28147-89-CryptoPro-Simple-C-ParamSet,
        id-Gost28147-89-CryptoPro-Simple-D-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-PublicKeyOctetString ::=
        OCTET STRING ( SIZE(64) )
    GostR3410-2001-PublicKey ::=
        BIT STRING ( SIZE(16..524) )
            -- Container for GostR3410-2001-PublicKeyOctetString
    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
            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-Simple-A-ParamSet |
                id-Gost28147-89-CryptoPro-Simple-B-ParamSet |
                id-Gost28147-89-CryptoPro-Simple-C-ParamSet |
                id-Gost28147-89-CryptoPro-Simple-D-ParamSet
            ) OPTIONAL
    }
GostR3410-2001-PublicKeyAlgorithms  ALGORITHM-IDENTIFIER ::= {
    { GostR3410-2001-PublicKeyParameters IDENTIFIED BY
                    id-GostR3410-2001 }
}
GostR3410-2001-CertificateSignatureAlgorithms
    ALGORITHM-IDENTIFIER ::= {
            { NULL IDENTIFIED BY
                    id-GostR3411-94-with-GostR3410-2001 } |
            { GostR3410-2001-PublicKeyParameters IDENTIFIED BY
                    id-GostR3411-94-with-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 which 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,
        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
        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)}
    ;
    GostR3410-2001-ParamSetParameters ::=
        SEQUENCE {
            a       INTEGER (
                        1
                    ..
                        115792089237316195423570985008687907853269984665640564039457584007913129639935
                        ),      -- 0 < a < p < 2^256
            b       INTEGER (
                        1
                    ..
                        115792089237316195423570985008687907853269984665640564039457584007913129639935
                        ),      -- 0 < b < p < 2^256
            p       INTEGER (
                        28948022309329048855892746252171976963317496166410141009864396001978282409985
                        ..
                        115792089237316195423570985008687907853269
```

```
8466564056403945758400791312963993 5

Popov,Kurepkin,Leontiev       Informational                    [Page 45]
```

```
Internet-Draft     Crypto-Pro cryptographic algorithms         July 2005


                        ),      -- 2^254 < p < 2^256
            q       INTEGER (
                        28948022309329048855892746252171976963317496166410141009864396001978282409985
                        ..
                        11579208923731619542357098500868790785326998466564056403945758400791312963993 5
                        ),      -- 2^254 < q < 2^256
            x       INTEGER (0
                    ..
                        11579208923731619542357098500868790785326998466564056403945758400791312963993 5
                        ),      -- 0 < x < p < 2^256
            y       INTEGER (0
                    ..
                        11579208923731619542357098500868790785326998466564056403945758400791312963993 5
                        )       -- 0 < y < p < 2^256
        }
  -- GOST R 34.10-2001 public key parameter set:
  -- OIDs for parameter sets are imported from GostR3410-2001-PKISy
ntax
    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
}
    }
    gostR3410-2001-TestParamSet
        AlgorithmIdentifier ::=
```

```
        {
            algorithm
                id-GostR3410-2001-TestParamSet,
            parameters
                GostR3410-2001-ParamSetParameters:{
                    a   7,
                    b   433088765467672769057659045956509319959421117944510395832529688420338495804
14,
```



```
                    -- 5FBFF498AA938CE739B8E022FBAFEF40
                    -- 563F6E6A3472FC2A514C0CE9DAE23B7E
                    p   578960446186580977117854925043439539266349923328202820197287920039565648210
41,
                    -- 800000000000000000000000000000000
                    -- 000000000000000000000000000000431
                    q   578960446186580977117854925043439539270829345837254506223809735921376310696
19,
                    -- 8000000000000000000000000000000150FE8A1892976154C59CFC193ACCF5B3
                    x   2,
                    y   401897405659037503335449422937059775635739389905545080690979365213431566280
                    --  8E2A8A0E65147D4BD6316030E16D19C
                    -- 85C97F0A9CA267122B96ABBCEA7E8FC8
                }
        }
    gostR3410-2001-CryptoPro-A-ParamSet
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3410-2001-CryptoPro-A-ParamSet,
            parameters
                GostR3410-2001-ParamSetParameters:{
                    a   115792089237316195423570985008687907853269984665640564039457584007913129639316,
                    -- -3 == p - 3
                    b   166,
                    -- a6
                    p   115792089237316195423570985008687907853269984665640564039457584007913129639319,
                    -- ffffffffffffffffffffffffffffffff
                    -- fffffffffffffffffffffffffffffd97
```

```
                    q   11579208923731619542357098500868790785307376290849924322537815580507906885032
3,
                        -- ffffffffffffffffffffffffffffffff
                        -- 6c611070995ad10045841b09b761b893
                    x   1,
                    y   64033881142927202683649881450434739859317602688849412888527458039088786386
12
                        -- 8d91e471e0989cda27df505a453f2b76
                        -- 35294f2ddf23e3b122acc99c9e9f1e14
                }
        }
    gostR3410-2001-CryptoPro-B-ParamSet
        AlgorithmIdentifier ::=
        {
            algorithm
```



```
                id-GostR3410-2001-CryptoPro-B-ParamSet,
            parameters
                GostR3410-2001-ParamSetParameters:{
                    a   57896044618658097711785492504343953926634992332820282019728792003956564823190,
                        -- -3 == p - 3
                    b   28091019353058090096996979000309560759124368558014865957655842872397301267595,
                        -- 3e1af419a269a5f866a7d3c25c3df80a
                        -- e979259373ff2b182f49d4ce7e1bbc8b
                    p   57896044618658097711785492504343953926634992332820282019728792003956564823193,
                        -- 8000000000000000000000000000000
                        -- 0000000000000000000000000000c99
                    q   57896044618658097711785492504343953927102133160255826820068844496087732066703,
                        -- 8000000000000000000000000000001
                        -- 5f700cfff1a624e5e497161bcc8a198f
                    x   1,
                    y   28792665814854611296992347458380284135028636778229113005756334730996303888124
                        -- 3fa8124359f96680b83d1c3eb2c070e5
                        -- c545c9858d03ecfb744bf8d717717efc
                }
        }
    gostR3410-2001-CryptoPro-C-ParamSet
```

```
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3410-2001-CryptoPro-C-ParamSet,
            parameters
                GostR3410-2001-ParamSetParameters:{
                    a   70390085352083305199547718019018437841079516
63004518047128434684370563350 2616,
                        -- -3 == p - 3
                    b   32858,
                        -- 805a
                    p   70390085352083305199547718019018437841079516
63004518047128434684370563350 2619,
                        -- 9b9f605f5a858107ab1ec85e6b41c8aa
                        -- cf846e86789051d37998f7b9022d759b
                    q   70390085352083305199547718019018437840920882
64716408103532260145835229839 6601,
                        -- 9b9f605f5a858107ab1ec85e6b41c8aa
                        -- 582ca3511eddfb74f02f3a6598980bb9
                    x   0,
                    y   29818893917731240733471273240314769927240550
81238369568914649526160456599 0247
```



```
                        -- 41ece55743711a8c3cbf3783cd08c0ee
                        -- 4d4dc440d4641a8f366e550dfdb3bb67
                }
        }
    gostR3410-2001-CryptoPro-ExA-ParamSet
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3410-2001-CryptoPro-XchA-ParamSet,
            parameters
                GostR3410-2001-ParamSetParameters:{
                    a   11579208923731619542357098500868790785326998
46656405640394575840079131296 39316,
                        -- -3 == p - 3
                    b   166,
                        -- a6
                    p   11579208923731619542357098500868790785326998
46656405640394575840079131296 39319,
                        -- ffffffffffffffffffffffffffffffff
```

```
                        -- ffffffffffffffffffffffffffffd97
                    q   11579208923731619542357098500868790785307376290849924322537815580507906885032
3,
                        -- ffffffffffffffffffffffffffffffff
                        -- 6c611070995ad10045841b09b761b893
                    x   1,
                    y   64033881142927202683649881450434773985931760268884941288852745803908878638612
                        -- 8d91e471e0989cda27df505a453f2b76
                        -- 35294f2ddf23e3b122acc99c9e9f1e14
                }
        }
    gostR3410-2001-CryptoPro-ExB-ParamSet
        AlgorithmIdentifier ::=
        {
            algorithm
                id-GostR3410-2001-CryptoPro-XchB-ParamSet,
            parameters
                GostR3410-2001-ParamSetParameters:{
                    a   70390085352083305199547718019018437841079516630045180471284346843705633502616,
                        -- -3 == p - 3
                    b   32858,
                        -- 805a
                    p   70390085352083305199547718019018437841079516630045180471284346843705633502619,
                        -- 9b9f605f5a858107ab1ec85e6b41c8aa
                        -- cf846e86789051d37998f7b9022d759b
                    q   70390085352083305199547718019018437840920882
```



```
26471640810353226014583522983966 01,
                        -- 9b9f605f5a858107ab1ec85e6b41c8aa
                        -- 582ca3511eddfb74f02f3a6598980bb9
                    x   0,
                    y   29818893917731240733471273240314769927240550812383695689146495261604565990247
                        -- 41ece55743711a8c3cbf3783cd08c0ee
                        -- 4d4dc440d4641a8f366e550dfdb3bb67
                }
        }
END -- GostR3410-2001-ParamSetSyntax
```

11 References

Normative references:

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

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

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

[GOSTR341194] "Information technology. Cryptographic Data Security. Hashing function.", GOST R 34.11-94, Gosudarstvennyi Standard of Russian Federation, Government Committee of the Russia for Standards, 1994. (In Russian);

[RFC 2119] Bradner, S., "Key Words for Use in RFCs to Indi- cateRequirement Levels", BCP 14, RFC 2119, March 1997.

[HMAC] H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

Informative references:

[Schneier95] B. Schneier, Applied cryptography, second edition, John Wiley & Sons, Inc., 1995;

[RFDSL] "Russian Federal Digital Signature Law", 10 Jan 2002 N1-FZ

[RFLLIC] "Russian Federal Law on Licensing of Selected Activity Categories", 08 Aug 2001 N 128-FZ

[CRYPTOLIC] "Russian Federal Goverment 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.

[TLS] The TLS Protocol Version 1.0. T. Dierks, C. Allen. January 1999, RFC 2246.

12 Acknowledgments

This document was created in accordance with "Russian Cryptographic Software Compatibility Agreement", signed by FGUE STC "Atlas", CRYPTO-PRO, Factor-TC, 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:

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.

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

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

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".

Author's Addresses

Vladimir Popov
CRYPTO-PRO
38, Obraztsova,
Moscow, 127018, Russian Federation
EMail: vpopov@cryptopro.ru

Igor Kurepkin
CRYPTO-PRO
38, Obraztsova,
Moscow, 127018, Russian Federation
EMail: kure@cryptopro.ru

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

Grigorij Chudov
CRYPTO-PRO
38, Obraztsova,
Moscow, 127018, Russian Federation
EMail: chudov@cryptopro.ru

Alexandr Afanasiev
Factor-TC
office 711, 14, Presnenskij val,
Moscow, 123557, Russian Federation
EMail: afa@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@stcnet.ru

Elena Minaeva
MD PREI
build 3, 6A, Vtoroj Troitskij per.,
Moscow, Russian Federation
EMail: evminaeva@mo.msk.ru

Serguei Murugov
R-Alpha
4/1, Raspletina,
Moscow, 123060, Russian Federation
EMail: msm@office.ru

Igori Ustinov
Cryptocom
office 239, 51, Leninskij prospekt,
Moscow, 119991, Russian Federation
EMail: igus@cryptocom.ru

Anatolij Erkin
SPRCIS (SPbRCZI)
1, Obrucheva,
St.Petersburg, 195220, Russian Federation
EMail: erkin@nevsky.net

Full Copyright Statement

Acknowledgment

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