Internet Draft
Vladimir Popov, CRYPTO-PRO
Igor Kurepkin, CRYPTO-PRO
Expires August 15, 2004
Serguei Leontiev, CRYPTO-PRO
Intended Category: Informational
February 15, 2004

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-00.txt>

Status of this Memo

Abstract

This document describes cryprographic algorithms and parameters, supplementary to 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

This document describes cryprographic algorithms, used in supplement to GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001 and GOST R 34.11-94, proposed by CRYPTO-PRO Company for "Russian Cryptographic Software Compatibility Agreement" community. GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001 and GOST R 34.11-94 are defined in corresponding national standards - [GOST28147], [GOSTR341094], [GOSTR34102001] and [GOSTR341194]. Their brief technical description in english can be found in [Schneier95].

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:

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 (I, K, D) - is D, encrypted with key K using GOST 28147-89 in "gammirovanie s obratnoj svyaziyu" (64-bit CFB) mode, and I as

initialization vector.

encryptOFB (I, K, D) - is D, encrypted with key K using GOST 28147-89 in "gammirovanie" (64-bit OFB) mode, and I as initialization vector.

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

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

2 Cipher parameters

[GOST28147] defines only the basic cryptographic operations, which can be used to encrypt or decrypt data. This document defines an additional cipher mode GOST 28147-89 CBC, and key meshing algorithm, which can be used to protect a symmetric key, when it is used to process large amounts of data.

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

2.1 GOST 28147-89 CBC mode

Algorithm GOST 28147-89 CBC mode is a block cipher with block chaining, based on GOST 28147-89 in ECB mode.

Before each plaintext block is encrypted, it is combined with the cipher text of the previous block by a bitwise exclusive OR operation. This ensures that even if the plaintext contains many identical blocks, they will each encrypt to a different cipher text

block. The initialization vector is combined with the first plaintext block by a bitwise exclusive OR operation before the block is encrypted.

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.

2.2 Key meshing algorithms

When there is a need to limit the amount of data, enciphered with the same key, several key meshing algorithms can be used.

```
id-Gost28147-89-None-KeyMeshing OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms keyMeshing(14) none(0) }
```

This is a zero key meshing algorithm - key is never changed.

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

This algorithm transforms the key and initialization vector every 1KB of plaintext data, using the following rule:

Let K[i] be the previous key, and S[i] be the previous initialization vector.

K[i+1] = decryptECB (K[i], C); S[i+1] = encryptECB (K[i+1],S[i])

Where C = {0x22720069L, 0x2304C964L, 0x96DB3A8DL, 0xC42AE946L,
0x94ACFE18L, 0x1207ED00L, 0xC2DC86C0L, 0x2BA94CEFL};

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(K,D) = P_GOSTR3411 (K,D)

5 Key establishment algorithms

Standards [GOSTR341094] and [GOSTR34102001] do not define any key establishment algorithms.

Section 5.7 specifies algorithms VKO GOST R 34.10-94 and VKO GOST R 34.10-2001, which can be used to export/import session key using a one-time exchange key (symmetric key, shared by sender and recipient), based on sender's private key and recipient public key, or vice versa.

Sections 5.1 and 5.2 describe how to create an exchange key from

sender's private key and recipient public key, or vice versa.

Section 5.3 describes how to create an export key from an exchange key or symmetric key.

Sections 5.4 and 5.5 describe, how a session key can be exported (encrypted) using export key, or using exchange key directly.

Section 5.6 describes, how to create session keys, using secret key and diversification data.

5.1 Creating exchange key using GOST R 34.10-94 keys

This algorithm creates an exchange key using sender's private key and recipient public key, or vice versa, using GOST R 34.10-94 public key algorithm and GOST R 34.11-94 hash function.

Exchange key EK is a 256-bit hash of 1024-bit Diffie-Hellman key K(x,y);

EK = gostR3411 (K(x,y))
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

Keys 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 Creating exchange key using GOST R 34.10-2001 keys

This algorithm creates an exchange key using sender's private key and recipient public key, or vice versa, using GOST R 34.10-2001 public key algorithm and GOST R 34.11-94 hash function.

Exchange key EK is a 256-bit hash of K(x,y,a);

EK(x,y,a) = gostR3411 (K(x,y,a))
K(x,y,a) = ((a*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)
a - synchrovector (64 bit)
P - base point on the elliptic curve (two 256-bit coordinates)
a*x - x multiplied by a as integers

x.P - a multiple point

Keys x and y MUST comply with [GOSTR34102001].

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

5.3 Generating export key from exchange key

Given a random 64-bit synchrovector A, and an exchange key K, produced by algorithms from sections 5.1 and 5.2 (or other shared symmetric key K), this algorithm creates an export key K(A), which can be used to export (encrypt) session key.

KA = K[8]. K[0]..K[8] are calculated with following algorithm:

```
K[0] = K;
K[i+1]  = encryptCFB (S[i], K[i], K[i])
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);

Here a[i,j] and k[i,j] are components of A and K[i] respectively:
K[i] = k[i,0]|k[i,1]|...|k[i,7] (k[i,j] - 32-bit integer)
A = a[0]|...|a[7] (a[i] - byte, a[i,0]..a[i,7] - it's bits)
```

5.4 Key export using export key

This algorithm exports session key SK using key K and random 64-bit synchrovector A. Outputs of this algorithm are 32-bit SK_mac and 256-bit SK_enc.

First, export key KA is created using algorithm, specified in 5.3, from the key K and vector A.

Then SK_mac is calculated: SK_mac = gost28147IMIT (A, KA, SK).

Then SK is encrypted in ECB mode, using key KA:

```
 SK_enc = encryptECB (KA, SK);
```

5.5 Key export using exchange key

This algorithm exports session key SK using exchange key K and random 64-bit synchrovector A. Outputs of this algorithm are 32-bit SK_mac and 256-bit SK_enc.

First, SK_mac is calculated: SK_mac = gost28147IMIT (A, K, SK).

Then SK is encrypted in ECB mode, using K for key:


```
 SK_enc = encryptECB (K, SK);
```

5.6 Key Diversification

This algorithm creates a session key SK, given 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 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 SV and 32-byte SRCKEY (B = SV|SRCKEY).

3) Algorithm from section 5.3 is used to create KA from key K and synchrovector SV, with two differences. Instead of S[i], vector (0,0,0,SV[i],ff,ff,ff,ff XOR SV[i]) is used, and during each encryption step, only 8 out of 32 GOST 28147-89 steps are done.

4) SK is calculated:
SK = encryptCFB (A, KA, SRCKEY).

5.7 VKO GOST R 34.10-94 and VKO GOST R 34.10-2001 algorithms. VKO GOST R 34.10-94 and VKO GOST R 34.10-2001 are key establishment algorithms for GOST R 34.10-94 and GOST R 34.10-2001 keys accordingly.

There are two modes they can be used in.

5.7.1 'Simple export' mode

Identifier for this mode:

```
id-Gost28147-89-None-KeyWrap OBJECT IDENTIFIER ::=
    { id-CryptoPro-algorithms keyWrap(13) none(0) }
```

The first step is calculating an exchange key, using algorithms, defined in sections 5.1 or 5.2, depending on key type.

Then, session key can be exported on this exchange key, using algorithm from section 5.5

5.7.2 'CryptoPro' mode

Identifier for this mode:

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

The first step is calculating an exchange key, using algorithms,

defined in sections 5.1 or 5.2, depending on key type.

Then, session key is exported on the export key (computed from this exchange key) using algorithm from section 5.4;

6 Algorithm parameters

Standards [GOST28147], [GOST341194], [GOSTR341094] and [GOSTR34102001] do not define specific values for algorithm parameters.

This document introduces the use of OIDs to specify algorithm parameters.

Identifiers and corresponding parameter values for all of the proposed parameter sets can be found in Appendix in the form of ASN.1 modules [X.660].

6.1 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 can be found in Appendix.

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

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

6.2 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 can be found in 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))
```

6.3 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 can be found in Appendix.

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

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



```
          }} OPTIONAL
    }

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

Where
p - modulus, prime number, 2^1023<p<2^1024;
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.

t - bit length of p;
x0 - seed;
c - used for p and q generation;
d - used for a generation.

6.4 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 can be found in Appendix.

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


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

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

7 Security Considerations

Parameter values for using cryptographic algorithms affect rigidity of information protection system. It is RECCOMENDED, that software

applications verify signature values, subject public keys and algorithm parameters to conform to [GOSTR34102001], [GOSTR341094] standards prior to their use.

The algorithm parameters proposed hereby and described in this document, have been analyzed by special certification laboratory of Scientific and Technical Centre "ATLAS" and by Centre of Certificational Investigations in appropriate levels of target_of_evaluation (TOE).

In case of different parameters usage, it is RECCOMENDED that they are to be examined by authorized agency with an approved methods of cryptographic analysis.

8 Appendix ASN.1 Modules

8.1 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,
        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 }
        AlgorithmIdentifier, ALGORITHM-IDENTIFIER
        FROM Cryptographic-Gost-Useful-Definitions
```

```
            cryptographic-Gost-Useful-Definitions
    ;
  -- GOST 28147-89 OID
    id-Gost28147-89 OBJECT IDENTIFIER ::=
        { id-CryptoPro-algorithms gost28147-89(21) }
  -- 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-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,
            macKey               Gost28147-89-MAC (SIZE (4))
        }
```

```
  -- GOST 28147-89 encryption algorithm parameters
    Gost28147-89-Parameters ::=
        SEQUENCE {
            encryptionParamSet
          OBJECT IDENTIFIER (
              id-Gost28147-89-TestParamSet | -- Only for tests use
              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
          ),
            iv                    Gost28147-89-IV
        }
    Gost28147-89-Algorithms ALGORITHM-IDENTIFIER ::= {
        { Gost28147-89-Parameters IDENTIFIED BY
              id-Gost28147-89 }
    }
END -- Gost28147-89-EncryptionSyntax
```

8.2 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,
        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, ALGORITHM-IDENTIFIER
        FROM Cryptographic-Gost-Useful-Definitions
             cryptographic-Gost-Useful-Definitions
    ;
  -- GOST 28147-89 Cryptographic Parameters Set:
  -- algorithm & parameters
  -- OID for Parameters Set imported from
  -- Gost28147-89-EncryptionSyntax
    Gost28147-89-ParamSetParameters ::=
            SEQUENCE {
            eUZ           Gost28147-89-UZ,
            mode          INTEGER {
                   gost28147-89-OFB(0),
                   gost28147-89-CFB(1),
                        cryptoPro-CBC(2)
               },
            shiftBits     INTEGER { gost28147-89-block(64) },
            keyWrap       AlgorithmIdentifier {{
                   Gost28147-89-KeyWrapAlgorithms
               }},
            keyMix        AlgorithmIdentifier {{
                   Gost28147-89-KeyMixAlgorithms
               }}
        }
    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-KeyMix  OBJECT IDENTIFIER ::=
        { id-CryptoPro-algorithms keyMix(14) cryptoPro(1) }
    id-Gost28147-89-None-KeyMix OBJECT IDENTIFIER ::=
        { id-CryptoPro-algorithms keyMix(14) none(0) }
    Gost28147-89-KeyMixAlgorithms  ALGORITHM-IDENTIFIER ::= {
        { NULL IDENTIFIED BY id-Gost28147-89-CryptoPro-KeyMix } |
        { NULL IDENTIFIED BY id-Gost28147-89-None-KeyMix }
    }
  -- GOST 28147-89 Cryptographic Parameters Set: values
    -- Test Parameters Set
    gost28147-89-TestParamSetAI
        AlgorithmIdentifier {{
            Gost28147-89-ParamSetAlgorithms
        }} ::=
        {
            algorithm
          id-Gost28147-89-TestParamSet,
            parameters
          Gost28147-89-ParamSetParameters:{
              eUZ   '4CDE389C2989EFB6FFEB56C55EC29B029875613B113F896
003970C798AA1D55DE210AD43375DB38EB42C77E7CD46CAFAD66A201F70F41EA4AB
```

```
03F22165B844D8'H,
```



```
                mode  gost28147-89-OFB,
            shiftBits  64,
                keyWrap
             { algorithm id-Gost28147-89-None-KeyWrap },
            keyMix
             { algorithm id-Gost28147-89-None-KeyMix }
        }
      }
    -- CryptoPro Parameters 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
        '93EEB31B67475ADA3E6A1D2F292C9C9588BD8170BA31D2AC1FD3F06E70
890B08A5C0E78642F245C2E65B2943FCA43459CB0FC8F104787F37DD15AEBD51966
6E4'H
    gost28147-89-CryptoPro-A-ParamSetAI
        AlgorithmIdentifier {{
            Gost28147-89-ParamSetAlgorithms
        }} ::=
        {
            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-KeyWrap },
            keyMix
             { algorithm id-Gost28147-89-CryptoPro-KeyMix }
      }
    }
```



```
    --
    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
        '80E7285041C57324B200C2AB1AADF6BE349B94985D265D1305D1AEC79C
B2BB3129731C7AE75A4142A38C07D9CFFFDF06DB346A6F686E80FD7619E985FE483
5EC'H
    gost28147-89-CryptoPro-B-ParamSetAI
        AlgorithmIdentifier {{
            Gost28147-89-ParamSetAlgorithms
        }} ::=
        {
            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-KeyWrap },
           keyMix
            { algorithm id-Gost28147-89-CryptoPro-KeyMix }
      }
    }
--
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
    '10838CA7B126D994C750BB602D0101859B4548DAD49D5EE205FA122FF2
A8240E483B97FC5E7233368FC9C651ECD7E5BBA96E6A4D7AEFF019661CAFC333B47
D78'H
gost28147-89-CryptoPro-C-ParamSetAI
    AlgorithmIdentifier {{
        Gost28147-89-ParamSetAlgorithms
    }} ::=
    {
        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-KeyWrap },
               keyMix
                { algorithm id-Gost28147-89-CryptoPro-KeyMix }
          }
        }
    --
    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
        'FB110831C6C5C00A23BE8F66A40C93F86CFAD21F4FE725EB5E60AE9002
5DBB2477A671DC9DD23A83E84B64C5D084574915994CB7BA33E9AD897FFD5231281
67E'H
    gost28147-89-CryptoPro-D-ParamSetAI
        AlgorithmIdentifier {{
            Gost28147-89-ParamSetAlgorithms
        }} ::=
        {
            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-KeyWrap },
            keyMix
             { algorithm id-Gost28147-89-CryptoPro-KeyMix }
      }
    }
--
gost28147-89-CryptoPro-Simple-A-ParamSetAI
    AlgorithmIdentifier {{
        Gost28147-89-ParamSetAlgorithms
    }} ::=
    {
        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 },
          keyMix
            { algorithm id-Gost28147-89-CryptoPro-KeyMix }
      }
    }
--
gost28147-89-CryptoPro-Simple-B-ParamSetAI
    AlgorithmIdentifier {{
        Gost28147-89-ParamSetAlgorithms
    }} ::=
```



```
    {
        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 },
          keyMix
            { algorithm id-Gost28147-89-CryptoPro-KeyMix }
```

```
          }
        }
    --
    gost28147-89-CryptoPro-Simple-C-ParamSetAI
        AlgorithmIdentifier {{
            Gost28147-89-ParamSetAlgorithms
        }} ::=
        {
            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 },
              keyMix
                { algorithm id-Gost28147-89-CryptoPro-KeyMix }
          }
        }
    --
    gost28147-89-CryptoPro-Simple-D-ParamSetAI
        AlgorithmIdentifier {{
            Gost28147-89-ParamSetAlgorithms
        }} ::=
        {
            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 },
```



```
              keyMix
                { algorithm id-Gost28147-89-CryptoPro-KeyMix }
          }
        }
END -- Gost28147-89-ParamSetSyntax
```

8.3 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,
        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
        AlgorithmIdentifier, ALGORITHM-IDENTIFIER
        FROM Cryptographic-Gost-Useful-Definitions
             cryptographic-Gost-Useful-Definitions
    ;
  -- GOST R 34.11-94 OID
    id-GostR3411-94 OBJECT IDENTIFIER ::=
        { id-CryptoPro-algorithms gostR3411-94(9) }
  -- GOST R 34.11-94 Cryptographic Parameters 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 Parameters & Algorithms
    GostR3411-94-DigestParameters ::=
        OBJECT IDENTIFIER (
          id-GostR3411-94-TestParamSet |     -- Only for tests use
          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
```

8.4 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,
        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, ALGORITHM-IDENTIFIER
        FROM Cryptographic-Gost-Useful-Definitions
```

```
            cryptographic-Gost-Useful-Definitions
    ;
  -- GOST R 34.11-94 Cryptographic Parameters Set:
  -- algorithm & parameters
  -- OID for Parameters Set imported from GostR3411-94-DigestSyntax
    GostR3411-94-ParamSetParameters ::=
        SEQUENCE {
            hUZ Gost28147-89-UZ,    -- S-Box for digest
            h0  GostR3411-94-Digest -- start 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 parameters set
    -- (GOST R 34.11-94 Annex A. Test vector)
    gostR3411TestParamSetAI AlgorithmIdentifier
     {{ GostR3411-94-ParamSetAlgorithms }} ::=
        {
            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
                  '4E5764D1AB8DCBBF941A7A4D2CD11010D6A057358D38F2F7
0F49D15AEA2F8D9462EE4309B3F4A6A218C698E3C17CE57E706B0966F7023C8B559
5BF2839B32ECC'H,
```



```
            h0   '0000000000000000000000000000000000000000000000000
0000000000000000'H
          }
        }
    -- CryptoPro parameters
    gostR3411CryptoProParamSetAI AlgorithmIdentifier
    {{ GostR3411-94-ParamSetAlgorithms }} ::=
        {
          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
              'A57477D14FFA66E354C7424A60ECB41982909D751D4FC90B3B122F5
47908A0AFD13E1A38C7B181C6E65605870325EBFE9C6DF86D2EABDE20BA893C92F8
D353BC'H,
            h0   '0000000000000000000000000000000000000000000000000
0000000000000000'H
```

```
          }
        }
END -- GostR3411-94-ParamSetSyntax
```

8.5 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,
        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
        AlgorithmIdentifier, ALGORITHM-IDENTIFIER
        FROM Cryptographic-Gost-Useful-Definitions
             cryptographic-Gost-Useful-Definitions
    ;
  -- GOST R 34.10-94 OIDs
    id-GostR3410-94 OBJECT IDENTIFIER ::=
        { id-CryptoPro-algorithms gostR3410-94(20) }
    id-GostR3411-94-with-GostR3410-94 OBJECT IDENTIFIER ::=
        { id-CryptoPro-algorithms
          gostR3411-94-with-gostR3410-94(4) }
  -- GOST R 34.10-94 Public Key Cryptographic Parameters 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 tests use
               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 tests use
              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 tests use
              id-GostR3411-94-CryptoProParamSet
          ),
            encryptionParamSet
          OBJECT IDENTIFIER (
              id-Gost28147-89-TestParamSet | -- Only for tests use
              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
```

8.6 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,
        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, ALGORITHM-IDENTIFIER
        FROM Cryptographic-Gost-Useful-Definitions
             cryptographic-Gost-Useful-Definitions
```

```
    ;
  -- GOST R 34.10-94 Public Key Cryptographic Parameters Set:
  -- algorithm & parameters
  -- OID for Parameters Set imported from GostR3410-94-PKISyntax
    GostR3410-94-ParamSetParameters ::=
        SEQUENCE {
            t    INTEGER (512 | 1024),  -- 512 - only for tests use
            p    INTEGER (
               167597599124282463744675312477573076593492072757404917221
544518046522050375919337210023428727086292846125398227331075635671923535149332124330420612576051
3
               ..
               134078079299425970995740249982058461274793658205923933777
235614437217640300735469768018742981669034276900318581864860508537538828119465699464336490060840
95
               |
               112355820928894744233081574424314045851123561183894160795
893800723582922378438101957942798326504710013200071174919620848536743605509010389058029644149671
327736104933390540928297688887250778808824658176845053128605523844176464039300921195694088017023
227094069177866436399967028711549822690522097706015140085773
               ..
               179769313486231590772930519078902473361797697894230657273
430081157732675805500963132708477322407536021120113879871393357658789768814416622492847430639474
124377767893424865485276302219601246094119453082952085005768838150682342462881473913110540827237
163350510684586298239947245938479716304835356329624224137215
               ),   -- 2^509 < p < 2^512 or 2^1020 < p < 2^1024
            q       INTEGER (
               289480223093290488558927462521719769633174961664101410098
643960019782824099 85
               ..
               115792089237316195423570985008687907853269984665640564039
457584007913129639935
               ),   -- 2^254 < q < 2^256
            a       INTEGER (
               2
               ..
               179769313486231590772930519078902473361797697894230657273
430081157732675805500963132708477322407536021120113879871393357658789768814416622492847430639474
124377767893424865485276302219601246094 609
```

```
411945308295208500576883815068234246288147391311054082723716335051068458629823994724593847971630483535632962422413721
4
                ),   -- 1 < a < p-1 < 2^1024-1
              validationAlgorithm
            AlgorithmIdentifier {{
                      GostR3410-94-ValidationAlgorithms
                }} OPTIONAL
        }
    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 {
            t    INTEGER (512 | 1024),  -- 512 - only for tests use
            x0   INTEGER (0 .. 65535),
            c    INTEGER (0 .. 65535),
            d    INTEGER (
               2
               ..
               1797693134862315907729305190789024733617976978942306572734300811577326758055009631327084773224075360211201138798713933576587897688144166224928474306394741243777678934248654852763022196012460941194530829520850057688381506823424628814739131105408272371633505106845862982399472459384797163048353563296242241372144
```

```
                )    -- 1 < d < p-1 < 2^1024-1
                OPTIONAL
        }
    GostR3410-94-ValidationBisParameters ::=
        SEQUENCE {
            t    INTEGER (512 | 1024),  -- 512 - only for tests use
            x0   INTEGER (0 .. 4294967295),
            c    INTEGER (0 .. 4294967295),
            d    INTEGER (
               2
               ..
               179769313486231590772930519078902473361797697894230657273
430081157732675805500963132708477322407536021120113879871393357658789768814416622492847430639474124377767893424865485276302219601246094119453082952085005768838150682342462881473913110540827237163350510684586298239947245938479716304835356329624224137214
                )    -- 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 Parameters sets
    -- GOST R 34.10-94 Tests parameters set
    -- (GOST R 34.10-94 Annex A. Test vector)
    gostR3410-94-TestParamSetAI
        AlgorithmIdentifier {{
            GostR3410-94-ParamSetAlgorithm
        }} ::=
        {
            algorithm
          id-GostR3410-94-TestParamSet,
            parameters
          GostR3410-94-ParamSetParameters:{
              t       512,
              p       12491554796616397392007291845361681019980789 08
```

```
4728846304013646795466302633346425772369277064638881858428879662416
2029257703157099684654914707531125817000 67,
              q       6900839799123747821852952871175357885746435622
1556536838757636132646301588781,
              a       8305821956779628193852750508811757244889982632
```



```
8218435214910357131733714685287987538317442674072307045274610623217
3266903443274617378695814257292977241346 8,
              validationAlgorithm {
                   algorithm
                    id-GostR3410-94-a,
                   parameters
                    GostR3410-94-ValidationParameters: {
                        t       512,
                        x0      24265,
                        c       29505,
                        d       2
                       }
                  }
          }
        }
    -- CryptoPro parameters
    gostR3410-94-CryptoPro-A-ParamSetAI
        AlgorithmIdentifier {{
            GostR3410-94-ParamSetAlgorithm
        }} ::=
        {
            algorithm
          id-GostR3410-94-CryptoPro-A-ParamSet,
            parameters
          GostR3410-94-ParamSetParameters:{
            t       1024,
            p       1270212482889324174659070427771764435257876535
0891653581281750726570503126098509849742318833348340118092599999512
0988934130659205614996724254121049274349357074920312769561451689224
1105793112488126102296785346384016935200132889950003622606842227508
1353230700451734163368500454106258697141688368678842537820383,
              q       6836319614495570078444416561182725289510217088
8761442055095051287550314083023,
              a       1009979067550553047720818155359252248698410825
7205345787482351587557714799052927277724415285269929879648335669968
2842027972896052747173175480590485607134746852141928680912561502802
```

```
22218564753919090265611636784727014501906679429093018544621639973087222173288983032319409735540321340097258832287685094674066396 2,
              validationAlgorithm {
                   algorithm
                    id-GostR3410-94-bBis,
                   parameters
                    GostR3410-94-ValidationBisParameters: {
                        t       1024,
                        x0      1376285941,
                        c       3996757427
                       }
               }
```



```
          }
        }
    --
    gostR3410-94-CryptoPro-B-ParamSetAI
        AlgorithmIdentifier {{
            GostR3410-94-ParamSetAlgorithm
        }} ::=
        {
            algorithm
          id-GostR3410-94-CryptoPro-B-ParamSet,
            parameters
          GostR3410-94-ParamSetParameters:{
              t       1024,
              p       139454871199115825601409655107690713107041707059928031797758001454375765357722984094124368522288239833039114681648076688236921220737322672160740747771700911134550432053804647694904686120113087816240740184800477047157336662926249423571248823968542221753660143391485680840520336859458494803187341288580489525163,
              q       79885141663410976897627118935756323747307951916507639758300472692338873533959,
              a       42941826148615804143873447737955502392672345968607143066798112994089471231420027060385216699563848719957657284814898909770759462613437669456364882730370838934791080835932647976778601915343474400961034231316672578686920482194932878633360203384797092684342247621055760235016132614780652761028509445403338652341,
              validationAlgorithm {
                   algorithm
                    id-GostR3410-94-bBis,
                   parameters
```

```
                    GostR3410-94-ValidationBisParameters: {
                        t       1024,
                        x0      1536654555,
                        c       1855361757,
                        d
   1440862938614001456765549029392820565478578022414617829967020177130599747551043947399151406115284791024439062735788342744854120601660303926203867703556828005895720381811489539897659442553756127180085030 6
                        }
                    }
          }
        }
    --
    gostR3410-94-CryptoPro-C-ParamSetAI
        AlgorithmIdentifier {{
            GostR3410-94-ParamSetAlgorithm
        }} ::=
        {
```



```
          algorithm
        id-GostR3410-94-CryptoPro-C-ParamSet,
          parameters
        GostR3410-94-ParamSetParameters:{
            t       1024,
            p       110624679233511963040518952417017040248586295481983138377419639629858439594897060895617022421062852556032786382467166554392976544029218447478930795186699928278807921929927011428546551433875806377110443534293554066712653034996277099320715774354228762128367184370370914135017194504580505029177050363451780493801,
            q       113468861199819350564868233378875198043267947776488510997961231672532899549103,
            a       816552717970881016017893191415300348226254405135358162468249467681876621283478212884286545844013955142622208772348502372286802227500950222482786620174449402169771648200835363982022980248926204808986993355080643323135297253322088194568951085155178100221003459370588291073071186553005962149936840737128710832 3,
            validationAlgorithm {
                 algorithm
                  id-GostR3410-94-bBis,
                 parameters
                  GostR3410-94-ValidationBisParameters: {
```

```
                         t       1024,
                         x0      113275885,
                         c       3037364845,
                         d       9175906676429839327
                        }
                   }
           }
         }
     --
     gostR3410-94-CryptoPro-D-ParamSetAI
         AlgorithmIdentifier {{
             GostR3410-94-ParamSetAlgorithm
         }} ::=
         {
             algorithm
           id-GostR3410-94-CryptoPro-D-ParamSet,
             parameters
           GostR3410-94-ParamSetParameters:{
               t       1024,
               p       905457649621929965904290958774625315611305608З
9073897669714048125244222625125560544746208559960915707867135849550
2367419155841859906278010664658095100957847139898194138208715964648
9144930534079207370788905204827306230388377677101736648382398574828
7878912864712014604743266126978496936655180738644364978932149,
               q       108988435796353506912374591498972192620190487557619582334771735390599299211593,
```



```
               a       756976611021707301782128757801610628085528380З
1095711588295742814192085325890416600170178598582163414003714687551
4127944005628789352666307543926770145985821033659831191739244732511
2254647122523868033159027077276687153434760863504720252982827271461
6901250506168582383843663310897774635410130339267237432548337,
               validationAlgorithm {
                    algorithm
                     id-GostR3410-94-bBis,
                    parameters
                     GostR3410-94-ValidationBisParameters: {
                         t       1024,
                         x0      333089693,
                         c       2699681355,
                         d
   6915887763901301481191744665240278894786443
```

```
2214275584246036624325 2
                        }
                    }
          }
        }
    --
    gostR3410-94-CryptoPro-XchA-ParamSetAI
        AlgorithmIdentifier {{
            GostR3410-94-ParamSetAlgorithm
        }} ::=
        {
            algorithm
          id-GostR3410-94-CryptoPro-XchA-ParamSet,
            parameters
          GostR3410-94-ParamSetParameters:{
              t       1024,
              p       14201174159756348119636828602231808974327613839524373876287257344192745939351271897363116607846760036084894662356762579528277471921224192907104613420838063639408451269182889400057152462544529576934935675272895683154177544176313938445719175509684710784659566254794231229333848392451433961472776068188060973423907659566254794231229333848392451433961472776068188060973423907,
              q       91771529896554605945588149018382750217296858393520724172743325725474374979801,
              a       13353181327272067343385951994831900121794237596784748689948235959936964252873471246159040332773182141032801252925387191478859899310331056774413619636480306472137782665689868646846327771015080940118260877020161532499046833293129492091277624113787803022435574660628397165937642683267426978088006163152816347588 7,
              validationAlgorithm {
                   algorithm
                    id-GostR3410-94-bBis,
                   parameters
                    GostR3410-94-ValidationBisParameters: {
```



```
                        t       1024,
                        x0      3495862036,
                        c       1177570399,
                        d
   3547889610240918895139647064772083281962391865341410582282334567466222018672580177997251216990526446086243776416033483110745 9
                    }
```

```
                }
          }
        }
    --
    gostR3410-94-CryptoPro-XchB-ParamSetAI
        AlgorithmIdentifier {{
            GostR3410-94-ParamSetAlgorithm
        }} ::=
        {
            algorithm
          id-GostR3410-94-CryptoPro-XchB-ParamSet,
            parameters
          GostR3410-94-ParamSetParameters:{
              t       1024,
              p       10289461266249948596765520743605303152179704999
989304888248413244847492302275847016799887100360467070487737728617 6
171227694098633153908956878412911010951269050334539386987129578346 7
257264868341720019662986056119366675242968236739708481517975203642 3
595736533689573920617698552845939650425308950460880671602694 33,
              q       9109671391802626916582318050603555673628769498
182593088388796888528164159519 9,
              a       8890864727828423151699995801875757891031463338
652579140051973659304813144068585706736982940794774449630665629150 5
503608252399443790027238674914599623086783222866197754399281674525 4
823298629859875357546628605173883785473616768576901778033580451144 0
773337196253842353291939447787366475282450998661787899244317 7
```



```
        }
```

```
    --
    gostR3410-94-CryptoPro-XchC-ParamSetAI
        AlgorithmIdentifier {{
            GostR3410-94-ParamSetAlgorithm
        }} ::=
        {
            algorithm
          id-GostR3410-94-CryptoPro-XchC-ParamSet,
            parameters
          GostR3410-94-ParamSetParameters:{
              t       1024,
              p       12469963669934775136071472657940644362034088613
950559892172484557299870737698999651480662364723992859320868822848
751165438350943327664722262594061556058045004094721182602772997756 3
540237169063044807971577164944778447000597419032457722226253269698
374446528353527293043937461065763833491510017159309241154995 49,
              q       67878761373365912343802950200656825271181294680
501479431146754294748422492761,
              a       44306184642975841824731350308098593268639906501
189417569952700748609973181426950235239623239110557450826919295792
878938752101867704718162325102751695310043185596483760265782782819 4
249605561893696586532551313719448313624777365346841011879674070984 0
825496997937556072234510670472108602597930996876319307290833 4,
              validationAlgorithm {
                   algorithm
                    id-GostR3410-94-bBis,
                   parameters
                    GostR3410-94-ValidationBisParameters: {
                        t       1024,
                        x0      371898640,
                        c       2482514131,
                        d
   39341170171309491894611690922945474002657559
065001688714824159421346618645269196467699 3
                       }
                  }
          }
        }
END -- GostR3410-94-ParamSetSyntax
```

8.7 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,
        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
        AlgorithmIdentifier, ALGORITHM-IDENTIFIER
        FROM Cryptographic-Gost-Useful-Definitions
             cryptographic-Gost-Useful-Definitions
    ;
  -- GOST R 34.10-2001 OIDs
    id-GostR3410-2001 OBJECT IDENTIFIER ::=
        { id-CryptoPro-algorithms gostR3410-2001(19) }
    id-GostR3411-94-with-GostR3410-2001 OBJECT IDENTIFIER ::=
        { id-CryptoPro-algorithms
          gostR3411-94-with-gostR3410-2001(3) }
  -- GOST R 34.10-2001 Public Key Cryptographic Parameters 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 tests use
              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 tests use
              id-GostR3411-94-CryptoProParamSet
          ),
            encryptionParamSet
          OBJECT IDENTIFIER (
              id-Gost28147-89-TestParamSet | -- Only for tests use
              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
```

8.8 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,
```

```
        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, ALGORITHM-IDENTIFIER
        FROM Cryptographic-Gost-Useful-Definitions
             cryptographic-Gost-Useful-Definitions
```



```
    ;
    GostR3410-2001-ParamSetParameters ::=
        SEQUENCE {
            a       INTEGER (
               1
                  ..
               115792089237316195423570985008687907853269984665640564039
457584007913129639935
               ),   -- 0 < a < p < 2^256
            b       INTEGER (
               1
                  ..
               115792089237316195423570985008687907853269984665640564039
457584007913129639935
               ),   -- 0 < b < p < 2^256
            p       INTEGER (
               289480223093290488558927462521719769633174961664101410098
643960019782824099 85
               ..
               115792089237316195423570985008687907853269984665640564039
457584007913129639935
               ),   -- 2^254 < p < 2^256
            q       INTEGER (
               289480223093290488558927462521719769633174961664101410098
643960019782824099 85
```

```
                ..
                115792089237316195423570985008687907853269984665640564039
457584007913129639935
                ),   -- 2^254 < q < 2^256
            x       INTEGER (0
                   ..
                115792089237316195423570985008687907853269984665640564039
457584007913129639935
                ),   -- 0 < x < p < 2^256
            y       INTEGER (0
                   ..
                115792089237316195423570985008687907853269984665640564039
457584007913129639935
                )    -- 0 < y < p < 2^256
        }
  -- GOST R 34.10-2001 Public Key Cryptographic Parameters Set:
  -- algorithm & parameters
  -- OID for Parameters Set imported from GostR3410-2001-PKISyntax
    GostR3410-2001-ParamSetAlgorithm ALGORITHM-IDENTIFIER ::= {
        { GostR3410-2001-ParamSetParameters IDENTIFIED BY
                id-GostR3410-2001-TestParamSet } |
        { GostR3410-2001-ParamSetParameters IDENTIFIED BY
                id-GostR3410-2001-CryptoPro-A-ParamSet  } |
```



```
        { GostR3410-2001-ParamSetParameters IDENTIFIED BY
                id-GostR3410-2001-CryptoPro-B-ParamSet  } |
        { GostR3410-2001-ParamSetParameters IDENTIFIED BY
                id-GostR3410-2001-CryptoPro-C-ParamSet  } |
        { GostR3410-2001-ParamSetParameters IDENTIFIED BY
                id-GostR3410-2001-CryptoPro-XchA-ParamSet  } |
        { GostR3410-2001-ParamSetParameters IDENTIFIED BY
                id-GostR3410-2001-CryptoPro-XchB-ParamSet  }
    }
    gostR3410-2001-TestParamSet
        AlgorithmIdentifier {{
            GostR3410-2001-ParamSetAlgorithm
        }} ::=
        {
            algorithm
          id-GostR3410-2001-TestParamSet,
            parameters
          GostR3410-2001-ParamSetParameters:{
```

```
            a   7,
            b   433088765467672769057659045956509319959421117944510
39583252968842033849580414,
                        -- 5FBFF498AA938CE739B8E022FBAFEF40563F6E6A
3472FC2A514C0CE9DAE23B7E
            p   578960446186580977117854925043439539266349923328202
82019728792003956564821041,
                -- 800000000000000000000000000000000000000000000000
00000000000000431
            q   578960446186580977117854925043439539270829345837254
50622380973592137631069619,
                        -- 8000000000000000000000000000000150FE8A18
92976154C59CFC193ACCF5B3
            x   2,
            y   401897405659037503335449422937059775635739389905545
08069097936521343156628O
                -- 8E2A8A0E65147D4BD6316030E16D19C85C97F0A9CA26712
2B96ABBCEA7E8FC8
          }
        }
    gostR3410-2001-CryptoPro-A-ParamSet
        AlgorithmIdentifier {{
            GostR3410-2001-ParamSetAlgorithm
        }} ::=
        {
            algorithm
          id-GostR3410-2001-CryptoPro-A-ParamSet,
            parameters
          GostR3410-2001-ParamSetParameters:{
            a   11579208923731619542357098500868790785326998466564
```



```
056403945758400791312963931 6,
             -- -3 == p - 3
            b   166,
                        -- a6
            p   11579208923731619542357098500868790785326998466564
056403945758400791312963931 9,
                -- ffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffd97
            q   11579208923731619542357098500868790785307376290849
924322537815580507906885032 3,
                        -- ffffffffffffffffffffffffffff6c611070
```

```
995ad10045841b09b761b893
              x   1,
              y   640338811429272026836498814504334739859317602688849412888527458039088786386 12
                  -- 8d91e471e0989cda27df505a453f2b7635294f2ddf23e3b122acc99c9e9f1e14
          }
        }
    gostR3410-2001-CryptoPro-B-ParamSet
        AlgorithmIdentifier {{
            GostR3410-2001-ParamSetAlgorithm
        }} ::=
        {
            algorithm
          id-GostR3410-2001-CryptoPro-B-ParamSet,
            parameters
          GostR3410-2001-ParamSetParameters:{
              a   57896044618658097711785492504343953926634992332820282019728792003956564823190,
               -- -3 == p - 3
              b   28091019353058090096996979000309560759124368558014865957655842872397301267595,
                  -- 3e1af419a269a5f866a7d3c25c3df80ae979259373ff2b182f49d4ce7e1bbc8b
              p   57896044618658097711785492504343953926634992332820282019728792003956564823193,
                  -- 8000000000000000000000000000000000000000000000000000000000000c99
              q   57896044618658097711785492504343953927102133160255826820068844496087732066703,
                        -- 8000000000000000000000000000000015f700cfff1a624e5e497161bcc8a198f
              x   1,
              y   28792665814854611296992347458380284135028636778229113005756334730996303888124
                  -- 3fa8124359f96680b83d1c3eb2c070e5c545c9858d03ecfb744bf8d717717efc
```



```
          }
        }
    gostR3410-2001-CryptoPro-C-ParamSet
        AlgorithmIdentifier {{
```

```
            GostR3410-2001-ParamSetAlgorithm
        }} ::=
        {
            algorithm
          id-GostR3410-2001-CryptoPro-C-ParamSet,
            parameters
          GostR3410-2001-ParamSetParameters:{
              a   70390085352083305199547718019018437841079516630045
180471284346843705633502616,
               -- -3 == p - 3
              b   32858,
                        -- 805a
              p   70390085352083305199547718019018437841079516630045
180471284346843705633502619,
                  -- 9b9f605f5a858107ab1ec85e6b41c8aacf846e86789051d
37998f7b9022d759b
              q   70390085352083305199547718019018437840920882647164
081035322601458352298396601,
                         -- 9b9f605f5a858107ab1ec85e6b41c8aa582ca351
1eddfb74f02f3a6598980bb9
              x   0,
              y   29818893917731240733471273240314769927240550812383
695689146495261604565990247
                  -- 41ece55743711a8c3cbf3783cd08c0ee4d4dc440d4641a8
f366e550dfdb3bb67
            }
        }
    gostR3410-2001-CryptoPro-ExA-ParamSet
        AlgorithmIdentifier {{
            GostR3410-2001-ParamSetAlgorithm
        }} ::=
        {
            algorithm
          id-GostR3410-2001-CryptoPro-XchA-ParamSet,
            parameters
          GostR3410-2001-ParamSetParameters:{
              a   11579208923731619542357098500868790785326998466564
056403945758400791312963931 6,
               -- -3 == p - 3
              b   166,
                        -- a6
              p   11579208923731619542357098500868790785326998466564
056403945758400791312963931 9,
                  -- ffffffffffffffffffffffffffffffffffffffffffffffff
```

```
fffffffffffffd97
              q   11579208923731619542357098500868790785307376290849
924322537815580507906885032 3,
                          -- ffffffffffffffffffffffffffffff6c611070
995ad10045841b09b761b893
              x   1,
              y   64033881142927202683649881450433473985931760268884
941288852745803908878638612
                  -- 8d91e471e0989cda27df505a453f2b7635294f2ddf23e3b
122acc99c9e9f1e14
          }
        }
    gostR3410-2001-CryptoPro-ExB-ParamSet
        AlgorithmIdentifier {{
            GostR3410-2001-ParamSetAlgorithm
        }} ::=
        {
            algorithm
          id-GostR3410-2001-CryptoPro-XchB-ParamSet,
            parameters
          GostR3410-2001-ParamSetParameters:{
              a   70390085352083305199547718019018437841079516630045
180471284346843705633502616,
               -- -3 == p - 3
              b   32858,
                  -- 805a
              p   70390085352083305199547718019018437841079516630045
180471284346843705633502619,
                  -- 9b9f605f5a858107ab1ec85e6b41c8aacf846e86789051d
37998f7b9022d759b
              q   70390085352083305199547718019018437840920882647164
081035322601458352298396601,
                          -- 9b9f605f5a858107ab1ec85e6b41c8aa582ca351
1eddfb74f02f3a6598980bb9
              x   0,
              y   29818893917731240733471273240314769927240550812383
695689146495261604565990247
                  -- 41ece55743711a8c3cbf3783cd08c0ee4d4dc440d4641a8
f366e550dfdb3bb67
          }
        }
END -- GostR3410-2001-ParamSetSyntax
```

9 References

[GOST28147] "Cryptographic Protection for Data Processing Sys-

tem", 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);

[GOSTR34102001] "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);

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

[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.

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

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

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

10 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 provided information about company products and solutions, and also for technical consulting in PKI.

RSA Security Russia and Demos Co Ltd for active collaboration and critical help in creation of this document.

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

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

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: aaaf@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