

Internet Draft

Expires October 5, 2005

Intended Category: Informational

Vladimir Popov, CRYPTO-PRO

Igor Kurepkin, CRYPTO-PRO

Serguei Leontiev, CRYPTO-PRO

April 5, 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-02.txt>](mailto:popov@cryptopro.ru)

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with [RFC 3668](http://www.rfc-editor.org/rfc/rfc3668).

This document is an Internet Draft and is subject to all provisions of [Section 10 of RFC2026](http://www.rfc-editor.org/rfc/rfc2026). Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet Drafts. Internet Drafts are draft documents valid for a maximum of 6 months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet Drafts as reference material or to cite them other than as a "work in progress".

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

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

Copyright (C) The Internet Society (2005). All Rights Reserved.

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

<u>1</u>	Introduction	<u>2</u>
<u>1.2</u>	Terminology.	<u>2</u>
<u>2</u>	Cipher modes and parameters.	<u>3</u>
<u>2.1</u>	GOST 28147-89 CBC mode	<u>3</u>

Internet-Draft

Crypto-Pro cryptographic algorithms

April 2005

2.2	GOST 28147-89 padding modes.	4
2.3	Key Meshing Algorithms	4
2.3.1	Null Key Meshing	4
2.3.2	CryptoPro Key Meshing.	4
3	HMAC_GOSTR3411	5
4	PRF_GOSTR3411.	5
5	Key Derivation Algorithms.	5
5.1	VKO GOST R 34.10-94.	5
5.2	VKO GOST R 34.10-2001.	6
6	Key Wrap algorithms.	6
6.1	GOST 28147-89 Key Wrap	6
6.2	GOST 28147-89 Key Unwrap.	7
6.3	CryptoPro Key Wrap	7
6.4	CryptoPro Key Unwrap	8
6.5	CryptoPro KEK Diversification Algorithm.	8
7	Secret Key Diversification	9
8	Algorithm parameters	9
8.1	Encryption algorithm parameters	9
8.2	Digest algorithm parameters.	11
8.3	GOST R 34.10-94 public key algorithm parameters	11
8.4	GOST R 34.10-2001 public key algorithm parameters.	12
9	Security Considerations.	13
10	Appendix ASN.1 Modules	14
11	References	49
12	Acknowledgments.	51
	Author's Address.	51
	Full Copyright Statement.	53

[1](#) Introduction

Russian cryptographic standards that define the algorithms GOST 28147-89 [[GOST28147](#)], GOST R 34.10-94 [[GOSTR341094](#)], GOST R 34.10-2001 [[GOSTR34102001](#)] 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 initialization 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

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

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 1KB of plaintext data.

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 $K[0] = K$, $IV[0] = IV$, $i =$

0. Let IV[0] be the value of the initialization vector after processing the first 1K block of data. Encryption or decryption of the next 1K data block 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 each 1K block of data:

- * the resulting initialization vector is stored as IV[i].
- * K[i+1] and IV0[i+1] are calculated
- * i is incremented.
- * Next block is encrypted or decrypted 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 [[GOSTR34102001](#)] 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 EK is a 256-bit hash of 1024-bit Diffie-Hellman key $K(x,y)$;

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

Keypairs x and y MUST comply with [\[GOSTR341094\]](#).

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

[5.2](#) VKO GOST R 34.10-2001

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

1. Let $K(x,y,UKM) = ((UKM*x) \pmod{q}) \cdot (y.P)$ (512 bit), where
x - sender's private key (256 bit)
x.P - sender's public key (512 bit)
y - recipient's private key (256 bit)

y.P - recipient's public key (512 bit)
UKM - 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 $K(x,y,UKM)$:
 $KEK(x,y,UKM) = \text{gostR3411}(K(x,y,UKM))$

Keypairs x and y MUST comply with [\[GOSTR34102001\]](#).

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

[7](#) Secret Key Diversification

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

1) 40-byte blob B is created from D by cloning it enough times to

fill all 40 bytes. For example, if D is 40-bytes long, $B = D$; If D is 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 ($B = \text{UKM}|\text{SRCKEY}$).

3) The algorithm from [section 6.5](#) is used to create $K(\text{UKM})$ from key K and UKM with two differences:

- * Instead of $S[i]$, vector $(0,0,0,\text{UKM}[i],\text{ff},\text{ff},\text{ff},\text{ff} \text{ XOR } \text{UKM}[i])$ is used.
- * During each encryption step, only 8 out of 32 GOST 28147-89 steps are done.

4) K_d is calculated:

$K_d = \text{encryptCFB}(\text{UKM}, K(\text{UKM}), \text{SRCKEY})$.

[8](#) 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 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]:

Popov,Kurepkin,Leontiev Informational [Page 11]

Internet-Draft Crypto-Pro cryptographic algorithms April 2005

```

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))

```

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 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 $aq \pmod p = 1$;
validationAlgorithm - constant p, q and a calculating algorithm.

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

[8.4](#) GOST R 34.10-2001 public key algorithm parameters

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

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

```
    id-GostR3410-2001-CryptoPro-XchB-ParamSet }  
}
```

Identifier values are in the Appendix.

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 [\[GOSTR34102001\]](#), [\[GOSTR341094\]](#) 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 [\[RFDSL\]](#), [\[RFLIC\]](#) and [\[CRYPTOLIC\]](#).

Use of different parameter sets is NOT RECCOMENDED. 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)
    }

```

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
```

Internet-Draft

Crypto-Pro cryptographic algorithms

April 2005

```
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-TransportLayerSecurity(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 |

```

Popov, Kurepkin, Leontiev

Informational

[Page 18]

Internet-Draft

Crypto-Pro cryptographic algorithms

April 2005

```

        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 test
ing 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
129731C7AE75A4142A38C07D9CFFFD06DB346A6F686E80FD7619E985FE4835EC'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

```

Internet-Draft

Crypto-Pro cryptographic algorithms

April 2005

```

-- 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 ::=
    {

```



```

-- 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 '0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000'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)

```

Internet-Draft

Crypto-Pro cryptographic algorithms

April 2005

```
        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-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-PublicKeyOcte
tString
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-PKISyntax
GostR3410-94-ParamSetParameters ::=
    SEQUENCE {
        t    INTEGER (512 | 1024), -- 512 - only for testing p

```

urposes

```

p    INTEGER (
        1675975991242824637446753124775730765934920
7275740491722154451804652205037591933721002342872708629284612539822
73310756356719235351493321243304206125760513
        ..
        1340780792994259709957402499820584612747936
5820592393377723561443721764030073546976801874298166903427690031858
186486050853753882811946569946433649006084095
        |
        1123558209288947442330815744243140458511235
6118389416079589380072358292237843810195794279832650471001320007117
4919620848536743605509010389058029644149671327736104933390540928297
6888872507788088246581768450531286055238441764640393009211956940880
1702322709406917786643639996702871154982269052209770601514008577
        ..
        1797693134862315907729305190789024733617976
9789423065727343008115773267580550096313270847732240753602112011387
9871393357658789768814416622492847430639474124377767893424865485276
3022196012460941194530829520850057688381506823424628814739131105408
27237163350510684586298239947245938479716304835356329624224137215
    ), -- 2^509 < p < 2^512 or 2^1020 < p

```

```

< 2^1024
    q      INTEGER (
                2894802230932904885589274625217197696331749
6166410141009864396001978282409985
                ..
                1157920892373161954235709850086879078532699
84665640564039457584007913129639935
            ),      -- 2^254 < q < 2^256
    a      INTEGER (
                2
                ..
                1797693134862315907729305190789024733617976
9789423065727343008115773267580550096313270847732240753602112011387
9871393357658789768814416622492847430639474124377767893424865485276
3022196012460941194530829520850057688381506823424628814739131105408
27237163350510684586298239947245938479716304835356329624224137214
            ),      -- 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
9789423065727343008115773267580550096313270847732240753602112011387
9871393357658789768814416622492847430639474124377767893424865485276
3022196012460941194530829520850057688381506823424628814739131105408
27237163350510684586298239947245938479716304835356329624224137214
                    )      -- 1 < d < p-1 < 2^1024-1
                    OPTIONAL
    }
GostR3410-94-ValidationBisParameters ::=
    SEQUENCE {
        x0    INTEGER (0 .. 4294967295),
        c     INTEGER (0 .. 4294967295),
        d     INTEGER (
                    2
                    ..
                    1797693134862315907729305190789024733617976
9789423065727343008115773267580550096313270847732240753602112011387
                    )      -- 1 < d < p-1 < 2^1024-1
                    OPTIONAL
    }

```

```

9871393357658789768814416622492847430639474124377767893424865485276
3022196012460941194530829520850057688381506823424628814739131105408
27237163350510684586298239947245938479716304835356329624224137214
                    )      -- 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      124915547966163973920072918453616810199
8078908472884630401364679546630263334642577236927706463888185842887
9662416202925770315709968465491470753112581700067,
        q      690083979912374782185295287117535788574
64356221556536838757636132646301588781,
        a      830582195677962819385275050881175724488
9982632821843521491035713173371468528798753831744267407230704527461
062321732669034432746173786958142572929772413468,
        validationAlgorithm {
          algorithm
            id-GostR3410-94-a,
          parameters
            GostR3410-94-ValidationParameters:
{
          x0      24265,
          c      29505,
          d      2
        }
      }
    }
  }
}

```

```

gostR3410-94-CryptoPro-A-ParamSetAI
  AlgorithmIdentifier ::=
  {
    algorithm
      id-GostR3410-94-CryptoPro-A-ParamSet,
    parameters
      GostR3410-94-ParamSetParameters:{
        t      1024,
        p      127021248288932417465907042777176443525
7876535089165358128175072657050312609850984974231883334834011809259
9999512098893413065920561499672425412104927434935707492031276956145
1689224110579311248812610229678534638401693520013288995000362260684
2227508135323070045173416336850045410625869714168836867788425378203
83,
        q      683631961449557007844441656118272528951
02170888761442055095051287550314083023,
        a      100997906755055304772081815535925224869
8410825720534578748235158755771479905292727772441528526992987964833
5669968284202797289605274717317548059048560713474685214192868091256
1502802222185647539190902656116367847270145019066794290930185446216
3997308722217328898303231940973554032134009725883228768509467406639
62,
        validationAlgorithm {
          algorithm
            id-GostR3410-94-bBis,
          parameters
            GostR3410-94-ValidationBisParameter
s: {
              x0      1376285941,
              c      3996757427
            }
          }
        }
      }
  }
--
gostR3410-94-CryptoPro-B-ParamSetAI
  AlgorithmIdentifier ::=
  {
    algorithm
      id-GostR3410-94-CryptoPro-B-ParamSet,
    parameters
      GostR3410-94-ParamSetParameters:{
        t      1024,
        p      139454871199115825601409655107690713107
0417070599280317977580014543757653577229840941243685222882398330391
1468164807668823692122073732267216074074777170091113455043205380464
7694904686120113087816240740184800477047157336662926249423571248823

```

9685422217536601433914856808405203368594584948031873412885804895251
63,

q 798851416634109768976271189357563237473
07951916507639758300472692338873533959,

a 429418261486158041438734477379555023926
7234596860714306679811299408947123142002706038521669956384871995765
7284814898909770759462613437669456364882730370838934791080835932647
9767786019153434744009610342313166725786869204821949328786333602033
8479709268434224762105576023501613261478065276102850944540333865234
1,

validationAlgorithm {
algorithm
id-GostR3410-94-bBis,
parameters
GostR3410-94-ValidationBisParameter

s: {
x0 1536654555,
c 1855361757,
d 14408629386140014567655
4902939282056547857802241461782996702017713059974755104394739915140
6115284791024439062735788342744854120601660303926203867703556828005
8957203818114895398976594425537561271800850306
}
}
}

--
gostR3410-94-CryptoPro-C-ParamSetAI
AlgorithmIdentifier ::=
{

algorithm
id-GostR3410-94-CryptoPro-C-ParamSet,
parameters
GostR3410-94-ParamSetParameters:{
t 1024,
p 110624679233511963040518952417017040248
5862954819831383774196396298584395948970608956170224210628525560327
8638246716655439297654402921844747893079518669992827880792192992701
1428546551433875806377110443534293554066712653034996277099320715774
3542287621283671843703709141350171945045805050291770503634517804938
01,

q 113468861199819350564868233378875198043
267947776488510997961231672532899549103,
a 816552717970881016017893191415300348226
2544051353358162468249467681876621283478212884286545844013955142622
208772348502372286802227500950224827866201744494021697716482008353

6398202298024892620480898699335508064332313529725332208819456895108
5155178100221003459370588291073071186553005962149936840737128710832

Internet-Draft

Crypto-Pro cryptographic algorithms

April 2005

3,

```
        validationAlgorithm {
            algorithm
                id-GostR3410-94-bBis,
            parameters
                GostR3410-94-ValidationBisParameter
s: {
            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
3056083907389766971404812524422262512556054474620855996091570786713
5849550236741915584185990627801066465809510095784713989819413820871
5964648914493053407920737078890520482730623038837767710173664838239
8574828787891286471201460474326612697849693665518073864436497893214
9,
        q      108988435796353506912374591498972192620
190487557619582334771735390599299211593,
        a      756976611021707301782128757801610628085
5283803109571158829574281419208532589041660017017859858216341400371
4687551412794400562878935266630754392677014598582103365983119173924
4732511225464712252386803315902707727668715343476086350472025298282
7271461690125050616858238384366331089777463541013033926723743254833
7,
```

```
        validationAlgorithm {
            algorithm
```

```

        id-GostR3410-94-bBis,
        parameters
        GostR3410-94-ValidationBisParameter
s: {
        x0      333089693,
        c      2699681355,
        d      69158877639013014811917
44665240278894786443822142755842460366243252
        }

```

Popov, Kurepkin, Leontiev

Informational

[Page 39]

Internet-Draft

Crypto-Pro cryptographic algorithms

April 2005

```

        }
    }
--
gostR3410-94-CryptoPro-XchA-ParamSetAI
AlgorithmIdentifier ::=
{
    algorithm
        id-GostR3410-94-CryptoPro-XchA-ParamSet,
    parameters
        GostR3410-94-ParamSetParameters:{
            t      1024,
            p      142011741597563481196368286022318089743
2761383952437387628725734419274593935127189736311660784676003608489
4662356762579528277471921224192907104613420838063639408451269182889
4000571524625445295769349356752728956831541775441763139384457191755
0968471078465956625479423122933384839245143396147277606818806097342
39,
            q      917715298965546059455881490183827502172
96858393520724172743325725474374979801,
            a      133531813272720673433859519948319001217
9423759678474868994823595993696425287347124615904033277318214103280
1252925387191478859899310331056774413619636480306472137782665689868
6468463277710150809401182608770201615324990468332931294920912776241
1378780302243557466062839716593764268326742697808800616315281634758
87,
        validationAlgorithm {
            algorithm
                id-GostR3410-94-bBis,
            parameters
                GostR3410-94-ValidationBisParameter
s: {

```


969728891533566058787317492748441827236576904274546146

```
    }
  }
}
--
gostR3410-94-CryptoPro-XchC-ParamSetAI
  AlgorithmIdentifier ::=
  {
    algorithm
      id-GostR3410-94-CryptoPro-XchC-ParamSet,
    parameters
      GostR3410-94-ParamSetParameters:{
        t      1024,
        p      124699636699347751360714726579406443620
3408861395055989217248455729987073769899965148066236472399285932086
8822848751165438350943327664722262594061556058045004094721182602772
9977563540237169063044807971577164944777844700059741903245772222625
3269698374446528353527293043937461065763833491510017159309241154995
49,
        q      678787613733659123438029502006568252711
81294680501479431146754294748422492761,
        a      443061846429758418247313503080985932686
```

```
3990650118941756995270074860997318142695023523962323911055745082691
9295792878938752101867704718162325102751695310043185596483760265782
7828194249605561893696586532551313719448313624777365346841011879674
0709840825496997937556072234510670472108602597930996876319307290833
4,
    validationAlgorithm {
      algorithm
        id-GostR3410-94-bBis,
      parameters
        GostR3410-94-ValidationBisParameter
s: {
        x0      371898640,
        c      2482514131,
        d      39341170171309491894611
6909229454740026575590650016887148241594213466186452691964676993
    }
  }
}
```

```
    }  
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-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-PublicKeyOc
tetString
    GostR3410-2001-PublicKeyParameters ::=
        SEQUENCE {
            publicKeyParamSet
            OBJECT IDENTIFIER (

```

```

testing purposes      id-GostR3410-2001-TestParamSet |    -- Only for
                      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
                ..
                    1157920892373161954235709850086879078532699
84665640564039457584007913129639935
                ),      -- 0 < a < p < 2^256
        b      INTEGER (
                    1
                ..
                    1157920892373161954235709850086879078532699
84665640564039457584007913129639935
                ),      -- 0 < b < p < 2^256
        p      INTEGER (
                    2894802230932904885589274625217197696331749
```

Internet-Draft

Crypto-Pro cryptographic algorithms

April 2005

```

        ..
        1157920892373161954235709850086879078532699
84665640564039457584007913129639935
        ),      -- 2^254 < p < 2^256
    q      INTEGER (
        2894802230932904885589274625217197696331749
6166410141009864396001978282409985
        ..
        1157920892373161954235709850086879078532699
84665640564039457584007913129639935
        ),      -- 2^254 < q < 2^256
    x      INTEGER (0
        ..
        1157920892373161954235709850086879078532699
84665640564039457584007913129639935
        ),      -- 0 < x < p < 2^256
    y      INTEGER (0
        ..
        1157920892373161954235709850086879078532699
84665640564039457584007913129639935
        )      -- 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 4330887654676727690576590459565093199594211
1794451039583252968842033849580414,
      -- 5FBFF498AA938CE739B8E022FBAFEF40563F6E6A
3472FC2A514C0CE9DAE23B7E
      p 5789604461865809771178549250434395392663499
2332820282019728792003956564821041,
      -- 800000000000000000000000000000000000000000
000000000000000000000000431
      q 5789604461865809771178549250434395392708293
4583725450622380973592137631069619,
      -- 80000000000000000000000000000000000000000150FE8A18
92976154C59CFC193ACCF5B3
      x 2,
      y 4018974056590375033354494229370597756357393
89905545080690979365213431566280
      -- 8E2A8A0E65147D4BD6316030E16D19C85C97F0A9
CA267122B96ABBCEA7E8FC8
    }
  }

```

```

gostR3410-2001-CryptoPro-A-ParamSet
  AlgorithmIdentifier ::=
  {
    algorithm
      id-GostR3410-2001-CryptoPro-A-ParamSet,
    parameters
      GostR3410-2001-ParamSetParameters:{
        a 1157920892373161954235709850086879078532699
84665640564039457584007913129639316,
        -- -3 == p - 3
        b 166,
        -- a6
        p 1157920892373161954235709850086879078532699

```



```

    }
}
gostR3410-2001-CryptoPro-C-ParamSet
  AlgorithmIdentifier ::=
  {
    algorithm
      id-GostR3410-2001-CryptoPro-C-ParamSet,
    parameters
      GostR3410-2001-ParamSetParameters:{
        a 7039008535208330519954771801901843784107951
6630045180471284346843705633502616,
        -- -3 == p - 3
        b 32858,
        -- 805a
        p 7039008535208330519954771801901843784107951
6630045180471284346843705633502619,
        -- 9b9f605f5a858107ab1ec85e6b41c8aacf846e86
789051d37998f7b9022d759b
        q 7039008535208330519954771801901843784092088
2647164081035322601458352298396601,
        -- 9b9f605f5a858107ab1ec85e6b41c8aa582ca351
1eddfb74f02f3a6598980bb9

```

```

        x 0,
        y 2981889391773124073347127324031476992724055
0812383695689146495261604565990247
        -- 41ece55743711a8c3cbf3783cd08c0ee4d4dc440
d4641a8f366e550dfdb3bb67
    }
}
gostR3410-2001-CryptoPro-ExA-ParamSet
  AlgorithmIdentifier ::=
  {
    algorithm
      id-GostR3410-2001-CryptoPro-XchA-ParamSet,
    parameters
      GostR3410-2001-ParamSetParameters:{
        a 1157920892373161954235709850086879078532699
84665640564039457584007913129639316,
        -- -3 == p - 3
        b 166,
        -- a6

```

```

        p 1157920892373161954235709850086879078532699
84665640564039457584007913129639319,
        -- ffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffd97
        q 1157920892373161954235709850086879078530737
62908499243225378155805079068850323,
        -- ffffffffffffffffffffffffffffffffff6c611070
995ad10045841b09b761b893
        x 1,
        y 6403388114292720268364988145043347398593176
0268884941288852745803908878638612
        -- 8d91e471e0989cda27df505a453f2b7635294f2d
df23e3b122acc99c9e9f1e14
    }
}
gostR3410-2001-CryptoPro-ExB-ParamSet
AlgorithmIdentifier ::=
{
    algorithm
        id-GostR3410-2001-CryptoPro-XchB-ParamSet,
    parameters
        GostR3410-2001-ParamSetParameters:{
            a 7039008535208330519954771801901843784107951
6630045180471284346843705633502616,
            -- -3 == p - 3
            b 32858,
            -- 805a
            p 7039008535208330519954771801901843784107951
6630045180471284346843705633502619,

```

```

        -- 9b9f605f5a858107ab1ec85e6b41c8aacf846e86
789051d37998f7b9022d759b
        q 7039008535208330519954771801901843784092088
2647164081035322601458352298396601,
        -- 9b9f605f5a858107ab1ec85e6b41c8aa582ca351
1eddfb74f02f3a6598980bb9
        x 0,
        y 2981889391773124073347127324031476992724055
0812383695689146495261604565990247
        -- 41ece55743711a8c3cbf3783cd08c0ee4d4dc440
d4641a8f366e550dfdb3bb67
    }

```


}
END -- GostR3410-2001-ParamSetSyntax

11 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);
- [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 Indicate Requirement 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](#).
- [RFDSL] "Russian Federal Digital Signature Law", 10 Jan 2002 N1-FZ
- [RFLLC] "Russian Federal Law on Licensing of Selected Activity Categories", 08 Aug 2001 N 128-FZ
- [CRYPTOLIC] "Russian Federal Government Regulation on Licensing of Selected Activity Categories in Cryptography Area", 23 Sep 2002 N 691

[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 Housley (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

Internet-Draft

Crypto-Pro cryptographic algorithms

April 2005

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, Obrucheve,
St.Petersburg, 195220, Russian Federation
EMail: erkin@nevsky.net

Full Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an

"AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.