Independent Submission
Request for Comments: 5832
Category: Informational
ISSN: 2070-1721

V. Dolmatov, Ed.
Cryptocom, Ltd.

March 2010

GOST R 34.10-2001: Digital Signature Algorithm

Abstract

This document is intended to be a source of information about the Russian Federal standard for digital signatures (GOST R 34.10-2001), which is one of the Russian cryptographic standard algorithms (called GOST algorithms). Recently, Russian cryptography is being used in Internet applications, and this document has been created as information for developers and users of GOST R 34.10-2001 for digital signature generation and verification.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5832.

Table of Contents

1. Introduction

1.1. General Information

1. GOST R 34.10-2001 [GOST3410] was developed by the Federal Agency for Government Communication and Information under the President of the Russian Federation with the participation of the All-Russia Scientific and Research Institute of Standardization.

   GOST R 34.10-2001 was submitted by Federal Agency for Government Communication and Information at President of the Russian Federation.

2. GOST R 34.10-2001 was accepted and activated by the Act 380-st of 12.09.2001 issued by the Government Committee of Russia for Standards.

3. GOST R 34.10-2001 was developed in accordance with the terminology and concepts of international standards ISO 2382-2:1976 "Data processing - Vocabulary - Part 2: Arithmetic and logic operations"; ISO/IEC 9796:1991 "Information technology -- Security techniques -- Digital signature schemes giving message recovery"; ISO/IEC 14888 "Information technology - Security techniques - Digital signatures with appendix"; and ISO/IEC 10118 "Information technology - Security techniques - Hash-functions".

4. GOST R 34.10-2001 replaces GOST R 34.10-94.

1.2. The Purpose of GOST R 34.10-2001

GOST R 34.10-2001 describes the generation and verification processes for digital signatures, based on operations with an elliptic curve points group, defined over a prime finite field.

GOST R 34.10-2001 has been developed to replace GOST R 34.10-94. Necessity for this development is caused by the need to increase digital signature security against unauthorized modification. Digital signature security is based on the complexity of discrete logarithm calculation in an elliptic curve points group and also on the security of the hash function used (according to [GOST3411]).

Terminologically and conceptually, GOST R 34.10-2001 is in accordance with international standards ISO 2382-2 [ISO2382-2], ISO/IEC 9796 [ISO9796-1991], ISO/IEC 14888 Parts 1-3 [ISO14888-1]-[ISO14888-3], and ISO/IEC 10118 Parts 1-4 [ISO10118-1]-[ISO10118-4].

Note: the main part of GOST R 34.10-2001 is supplemented with three appendixes:

"Extra Terms in the Digital Signature Area" (Appendix A of this memo);

"Test Examples" (Section 7 of this memo);

"A Bibliography in the Digital Signature Area" (Section 9.2 of this memo).

2. Applicability

GOST R 34.10-2001 defines an electronic digital signature (or simply digital signature) scheme, digital signature generation and verification processes for a given message (document), meant for transmission via insecure public telecommunication channels in data processing systems of different purposes.

Use of a digital signature based on GOST R 34.10-2001 makes transmitted messages more resistant to forgery and loss of integrity, in comparison with the digital signature scheme prescribed by the previous standard.

GOST R 34.10-2001 is obligatory to use in the Russian Federation in all data processing systems providing public services.

3. Definitions and Notations

3.1. Definitions

The following terms are used in the standard:

Appendix: Bit string, formed by a digital signature and by the arbitrary text field [ISO14888-1].

Signature key: Element of secret data, specific to the subject and used only by this subject during the signature generation process [ISO14888-1].

Verification key: Element of data mathematically linked to the signature key data element, used by the verifier during the digital signature verification process [ISO14888-1].

Domain parameter: Element of data that is common for all the subjects of the digital signature scheme, known or accessible to all the subjects [ISO14888-1].

Signed message: A set of data elements, which consists of the message and the appendix, which is a part of the message.

Pseudo-random number sequence: A sequence of numbers, which is obtained during some arithmetic (calculation) process, used in a specific case instead of a true random number sequence [ISO2382-2].

Random number sequence: A sequence of numbers none of which can be predicted (calculated) using only the preceding numbers of the same sequence [ISO2382-2].

Verification process: A process that uses the signed message, the verification key, and the digital signature scheme parameters as initial data and that gives the conclusion about digital signature validity or invalidity as a result [ISO14888-1].

Signature generation process: A process that uses the message, the signature key, and the digital signature scheme parameters as initial data and that generates the digital signature as the result [ISO14888-1].

Witness: Element of data (resulting from the verification process) that states to the verifier whether the digital signature is valid or invalid [ISO148881-1]).

Random number: A number chosen from the definite number set in such a way that every number from the set can be chosen with equal probability [ISO2382-2].

Message: String of bits of a limited length [ISO9796-1991].

Hash code: String of bits that is a result of the hash function [ISO148881-1].

Hash function: The function, mapping bit strings onto bit strings of fixed length observing the following properties:

1) it is difficult to calculate the input data, that is the pre-image of the given function value;

2) it is difficult to find another input data that is the pre-image of the same function value as is the given input data;

3) it is difficult to find a pair of different input data, producing the same hash function value.

Note: Property 1 in the context of the digital signature area means that it is impossible to recover the initial message using the digital signature; property 2 means that it is difficult to find another (falsified) message that produces the same digital signature

as a given message; property 3 means that it is difficult to find some pair of different messages, which both produce the same signature.

(Electronic) Digital signature: String of bits obtained as a result of the signature generation process. This string has an internal structure, depending on the specific signature generation mechanism.

Note: In GOST R 34.10-2001 terms, "Digital signature" and "Electronic digital signature" are synonymous to save terminological succession to native legal documents currently in force and scientific publications.

3.2. Notations

In GOST R 34.10-2001, the following notations are used:

V256 - set of all binary vectors of a 256-bit length

V_all - set of all binary vectors of an arbitrary finite length

Z - set of all integers

p - prime number, p > 3

GF(p) - finite prime field represented by a set of integers {0, 1, ..., p - 1}

b (mod p) - minimal non-negative number, congruent to b modulo p

M - user’s message, M belongs to V_all

(H1 || H2) - concatenation of two binary vectors

a,b - elliptic curve coefficients

m - points of the elliptic curve group order

q - subgroup order of group of points of the elliptic curve

O - zero point of the elliptic curve

P - elliptic curve point of order q

d - integer - a signature key

Q - elliptic curve point - a verification key

^ - the power operator

/= - non-equality

sqrt - square root

zeta - digital signature for the message M

4. General Statements

A commonly accepted digital signature scheme (model) (see Section 6 of [ISO/IEC14888-1]) consists of three processes:

- generation of a pair of keys (for signature generation and for signature verification);
- signature generation;
- signature verification.

In GOST R 34.10-2001, a process for generating a pair of keys (for signature and verification) is not defined. Characteristics and ways of the process realization are defined by involved subjects, who determine corresponding parameters by their agreement.

The digital signature mechanism is defined by the realization of two main processes (see Section 7):

- signature generation (see Section 6.1) and
- signature verification (see Section 6.2).

The digital signature is meant for the authentication of the signatory of the electronic message. Besides, digital signature usage gives an opportunity to provide the following properties during signed message transmission:

- realization of control of the transmitted signed message integrity,
- proof of the authorship of the signatory of the message,
- protection of the message against possible forgery.

A schematic representation of the signed message is shown in Figure 1.

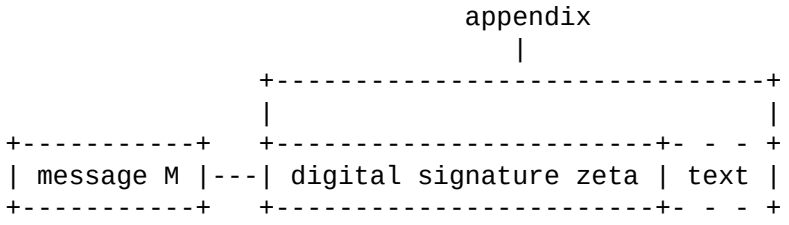

Figure 1: Signed message scheme

The field "digital signature" is supplemented by the field "text" (see Figure 1), that can contain, for example, identifiers of the signatory of the message and/or time label.

The digital signature scheme determined in GOST R 34.10-2001 must be implemented using operations of the elliptic curve points group, defined over a finite prime field, and also with the use of hash function.

The cryptographic security of the digital signature scheme is based on the complexity of solving the problem of the calculation of the discrete logarithm in the elliptic curve points group and also on the security of the hash function used. The hash function calculation algorithm is determined in [GOST3411].

The digital signature scheme parameters needed for signature generation and verification are determined in Section 5.2.

GOST R 34.10-2001 does not determine the process of generating parameters needed for the digital signature scheme. Possible sets of these parameters are defined, for example, in [RFC4357].

The digital signature represented as a binary vector of a 512-bit length must be calculated using a definite set of rules, as stated in Section 6.1.

The digital signature of the received message is accepted or denied in accordance with the set of rules, as stated in Section 6.2.

5. Mathematical Conventions

To define a digital signature scheme, it is necessary to describe basic mathematical objects used in the signature generation and verification processes. This section lays out basic mathematical definitions and requirements for the parameters of the digital signature scheme.

5.1. Mathematical Definitions

Suppose a prime number p > 3 is given. Then, an elliptic curve E, defined over a finite prime field GF(p), is the set of number pairs (x,y), x, y belong to Fp, satisfying the identity:

$$y^2 = x^3 + a*x + b \pmod{p}, \quad (1)$$

where a, b belong to GF(p) and 4*a^3 + 27*b^2 is not congruent to zero modulo p.

An invariant of the elliptic curve is the value J(E), satisfying the equality:

$$J(E) = 1728 * \frac{4*a^3}{4*a^3+27*b^2} \pmod{p} \quad (2)$$

Elliptic curve E coefficients a,b are defined in the following way using the invariant J(E):

$$\left| \begin{array}{l} a=3*k \pmod{p} \\ b=2*k \pmod{p}, \text{ where } k = \frac{J(E)}{1728 - J(E)} \pmod{p}, \; J(E) \neq 0 \text{ or } 1728 \end{array} \right. \quad (3)$$

The pairs (x,y) satisfying the identity (1) are called the elliptic curve E points; x and y are called x- and y-coordinates of the point, correspondingly.

We will denote elliptic curve points as Q(x,y) or just Q. Two elliptic curve points are equal if their x- and y-coordinates are equal.

On the set of all elliptic curve E points, we will define the addition operation, denoted by "+". For two arbitrary elliptic curve E points Q1 (x1, y1) and Q2 (x2, y2), we will consider several variants.

Suppose coordinates of points Q1 and Q2 satisfy the condition x1 /= x2. In this case, their sum is defined as a point Q3 (x3,y3), with coordinates defined by congruencies:

$$\left| \begin{array}{l} x3=\lambda^2-x1-x2 \pmod{p}, \\ y3=\lambda*(x1-x3)-y1 \pmod{p}, \end{array} \right. \text{ where } \lambda = \frac{y1-y2}{x1-x2} \pmod{p}. \quad (4)$$

If x1 = x2 and y1 = y2 /= 0, then we will define point Q3 coordinates in the following way:

```
| x3=lambda^2-x1*2 (mod p),                    3*x1^2+a
|                              where lambda= --------- (mod p)   (5)
| y3=lambda*(x1-x3)-y1 (mod p),                 y1*2
```

If x1 = x2 and y1 = - y2 (mod p), then the sum of points Q1 and Q2 is called a zero point O, without determination of its x- and y-coordinates. In this case, point Q2 is called a negative of point Q1. For the zero point, the equalities hold:

```
O+Q=Q+O=Q,                                                    (6)
```

where Q is an arbitrary point of elliptic curve E.

A set of all points of elliptic curve E, including zero point, forms a finite abelian (commutative) group of order m regarding the introduced addition operation. For m, the following inequalities hold:

```
p + 1 - 2*sqrt(p) =< m =< p + 1 + 2*sqrt(p).                  (7)
```

The point Q is called a point of multiplicity k, or just a multiple point of the elliptic curve E, if for some point P the following equality holds:

```
Q = P + ... + P = k*P.                                        (8)
    -----+-----
         k
```

5.2. Digital Signature Parameters

The digital signature parameters are:

- prime number p is an elliptic curve modulus, satisfying the inequality p > 2^255. The upper bound for this number must be determined for the specific realization of the digital signature scheme;

- elliptic curve E, defined by its invariant J(E) or by coefficients a, b belonging to GF(p).

- integer m is an elliptic curve E points group order;

- prime number q is an order of a cyclic subgroup of the elliptic curve E points group, which satisfies the following conditions:

```
| m = nq, n belongs to Z , n>=1
|                                                          (9)
| 2^254 < q < 2^256
```

- point P /= O of an elliptic curve E, with coordinates (x_p, y_p), satisfying the equality q*P=O.

- hash function h(.):V_all -> V256, which maps the messages represented as binary vectors of arbitrary finite length onto binary vectors of a 256-bit length. The hash function is determined in [GOST3411].

Every user of the digital signature scheme must have its personal keys:

- signature key, which is an integer d, satisfying the inequality 0 < d < q;

- verification key, which is an elliptic curve point Q with coordinates (x_q, y_q), satisfying the equality d*P=Q.

The previously introduced digital signature parameters must satisfy the following requirements:

- it is necessary that the condition p^t/= 1 (mod q) holds for all integers t = 1, 2, ... B where B satisfies the inequality B >= 31;

- it is necessary that the inequality m /= p holds;

- the curve invariant must satisfy the condition J(E) /= 0, 1728.

5.3. Binary Vectors

To determine the digital signature generation and verification processes, it is necessary to map the set of integers onto the set of binary vectors of a 256-bit length.

Consider the following binary vector of a 256-bit length where low-order bits are placed on the right, and high-order ones are placed on the left:

```
H = (alpha[255], ... , alpha[0]), H belongs to V256              (10)
```

where alpha[i], i = 0, ... , 255 are equal to 1 or to 0. We will say that the number alpha belonging to Z is mapped onto the binary vector h, if the equality holds:

alpha = alpha[0]*2^0 + alpha[1]*2^1 + ... + alpha[255]*2^255 (11)

For two binary vectors H1 and H2, which correspond to integers alpha and beta, we define a concatenation (union) operation in the following way. If:

H1 = (alpha[255], ... , alpha[0]),
(12)
H2 = (beta[255], ..., beta[0]),

then their union is

H1||H2 = (alpha[255], ... , alpha[0], beta[255], ..., beta[0]) (13)

that is a binary vector of 512-bit length, consisting of coefficients of the vectors H1 and H2.

On the other hand, the introduced formulae define a way to divide a binary vector H of 512-bit length into two binary vectors of 256-bit length, where H is the concatenation of the two.

6. Main Processes

In this section, the digital signature generation and verification processes of user's message are defined.

For the realization of the processes, it is necessary that all users know the digital signature scheme parameters, which satisfy the requirements of Section 5.2.

Besides, every user must have the signature key d and the verification key Q(x[q], y[q]), which also must satisfy the requirements of Section 5.2.

6.1. Digital Signature Generation Process

It is necessary to perform the following actions (steps) according to Algorithm I to obtain the digital signature for the message M belonging to V_all:

Step 1 - calculate the message hash code M: H = h(M). (14)

Step 2 - calculate an integer alpha, binary representation of which is the vector H, and determine e = alpha (mod q). (15)

If e = 0, then assign e = 1.

Step 3 - generate a random (pseudorandom) integer k, satisfying the inequality:

$$0 < k < q. \quad (16)$$

Step 4 - calculate the elliptic curve point C = k*P and determine if:

$$r = x_C \pmod q, \quad (17)$$

where x_C is x-coordinate of the point C. If r = 0, return to step 3.

Step 5 - calculate the value:

$$s = (r*d + k*e) \pmod q. \quad (18)$$

If s = 0, return to step 3.

Step 6 - calculate the binary vectors R and S, corresponding to r and s, and determine the digital signature zeta = (R || S) as a concatenation of these two binary vectors.

The initial data of this process are the signature key d and the message M to be signed. The output result is the digital signature zeta.

6.2. Digital Signature Verification

To verify digital signatures for the received message M belonging to V_all, it is necessary to perform the following actions (steps) according to Algorithm II:

Step 1 - calculate the integers r and s using the received signature zeta. If the inequalities 0 < r < q, 0 < s < q hold, go to the next step. Otherwise, the signature is invalid.

Step 2 - calculate the hash code of the received message M:

$$H = h(M). \quad (19)$$

Step 3 - calculate the integer alpha, the binary representation of which is the vector H, and determine if:

$$e = alpha \pmod q. \quad (20)$$

If e = 0, then assign e = 1.

Step 4 - calculate the value v = e^(-1) (mod q). (21)

Step 5 - calculate the values:

z1 = s*v (mod q), z2 = -r*v (mod q). (22)

Step 6 - calculate the elliptic curve point C = z1*P + z2*Q and determine if:

R = x_C (mod q), (23)

where x_C is x-coordinate of the point.

Step 7 - if the equality R = r holds, then the signature is accepted. Otherwise, the signature is invalid.

The input data of the process are the signed message M, the digital signature zeta, and the verification key Q. The output result is the witness of the signature validity or invalidity.

7. Test Examples (Appendix to GOST R 34.10-2001)

This section is included in GOST R 34.10-2001 as a reference appendix but is not officially mentioned as a part of the standard.

The values given here for the parameters p, a, b, m, q, P, the signature key d, and the verification key Q are recommended only for testing the correctness of actual realizations of the algorithms described in GOST R 34.10-2001.

All numerical values are introduced in decimal and hexadecimal notations. The numbers beginning with 0x are in hexadecimal notation. The symbol "\\" denotes a hyphenation of a number to the next line. For example, the notation:

```
12345\\
67890

0x499602D2
```

represents 1234567890 in decimal and hexadecimal number systems, respectively.

7.1. The Digital Signature Scheme Parameters

The following parameters must be used for the digital signature generation and verification (see Section 5.2).

7.1.1. Elliptic Curve Modulus

The following value is assigned to parameter p in this example:

p= 57896044618658097711785492504343953926\\
634992332820282019728792003956564821041

p = 0x8000000000000000000000000000\\
00000000000000000000000000000000000431

7.1.2. Elliptic Curve Coefficients

Parameters a and b take the following values in this example:

a = 7
a = 0x7

b = 43308876546767276905765904595650931995\\
942111794451039583252968842033849580414

b = 0x5FBFF498AA938CE739B8E022FBAFEF40563\\
F6E6A3472FC2A514C0CE9DAE23B7E

7.1.3. Elliptic Curve Points Group Order

Parameter m takes the following value in this example:

m = 57896044618658097711785492504343953926\\
708293458372545062238097359213763106961 9

m = 0x8000000000000000000000000000\\
00150FE8A1892976154C59CFC193ACCF5B3

7.1.4. Order of Cyclic Subgroup of Elliptic Curve Points Group

Parameter q takes the following value in this example:

q = 57896044618658097711785492504343953926\\
708293458372545062238097359213763106961 9

q = 0x8000000000000000000000000000001\\
50FE8A1892976154C59CFC193ACCF5B3

7.1.5. Elliptic Curve Point Coordinates

Point P coordinates take the following values in this example:

x_p = 2
x_p = 0x2

y_p = 40189740565390375033354494229370597\\
75635739389905545080690979365213431566280

y_p = 0x8E2A8A0E65147D4BD6316030E16D19\\
C85C97F0A9CA267122B96ABBCEA7E8FC8

7.1.6. Signature Key

It is supposed, in this example, that the user has the following signature key d:

d = 55441196065363246126355624130324183l\\
96576709222340016572108097750006097525544

d = 0x7A929ADE789BB9BE10ED359DD39A72C\\
11B60961F49397EEE1D19CE9891EC3B28

7.1.7. Verification Key

It is supposed, in this example, that the user has the verification key Q with the following coordinate values:

x_q = 57520216126176808443631405023338071\\
17663010490631363218289674134220660485940 3

x_q = 0x7F2B49E270DB6D90D8595BEC458B5\\
0C58585BA1D4E9B788F6689DBD8E56FD80B

y_q = 17614944419213781543809391949654080\\
03194266204536363926070984785943828676399 4

y_q = 0x26F1B489D6701DD185C8413A977B3\\
CBBAF64D1C593D26627DFFB101A87FF77DA

7.2. Digital Signature Process (Algorithm I)

Suppose that after steps 1-3, according to Algorithm I (Section 6.1), are performed, the following numerical values are obtained:

e = 20798893674476452017134061561508270l3\\
06371425153796532899526172526614688724 21

e = 0x2DFBC1B372D89A1188C09C52E0EE\\
C61FCE52032AB1022E8E67ECE6672B043EE5

k = 5385413767734846373140384114799661920\\
4150400343430202071296083852889319623339 5

k = 0x77105C9B20BCD3122823C8CF6FCC\\
7B956DE33814E95B7FE64FED924594DCEAB3

And the multiple point C = k * P has the coordinates:

x_C = 29700980915817952874371204983938256 9\\
9042275210799431965163268798205921093339 5

x_C = 0x41AA28D2F1AB148280CD9ED56FED\\
A41974053554A42767B83AD043FD39DC0493

y[C] = 32842535278684663477094665322517084 5\\
0680472103245454326813285455653927406091 0

y[C] = 0x489C375A9941A3049E33B34361DD\\
204172AD98C3E5916DE27695D22A61FAE46E

Parameter r = x_C(mod q) takes the value:

r = 29700980915817952874371204983938256 9\\
9042275210799431965163268798205921093339 5

r = 0x41AA28D2F1AB148280CD9ED56FED\\
A41974053554A42767B83AD043FD39DC0493

Parameter s = (r*d + k*e)(mod q) takes the value:

s = 57497340027008465417892531001914703\\
8455227042649098563933718999175515839552

s = 0x1456C64BA4642A1653C235A98A602\\
49BCD6D3F746B631DF928014F6C5BF9C40

7.3. Verification Process of Digital Signature (Algorithm II)

Suppose that after steps 1-3, according to Algorithm II (Section 6.2), are performed, the following numerical value is obtained:

e = 2079889367447645201713406156150827013\\
0637142515379653289952617252661468872421

```
e = 0x2DFBC1B372D89A1188C09C52E0EE\\
C61FCE52032AB1022E8E67ECE6672B043EE5

And the parameter v = e^(-1) (mod q) takes the value:

v = 176866836059344686773017138249002685\\
627468830806754967152880365724311457189 78

v = 0x271A4EE429F84EBC423E388964555BB\\
29D3BA53C7BF945E5FAC8F381706354C2

The parameters z1 = s*v(mod q) and z2 = -r*v(mod q) take the values:

z1 = 37699167500901938556841057293512656 1\\
088413451904919426193045324127437209997 59

z1 = 0x5358F8FFB38F7C09ABC782A2DF2A\\
3927DA4077D07205F763682F3A76C9019B4F

z2 = 14171998427343472112515917969500765 7\\
692466558389728621144999326533336710922 1

z2 = 0x3221B4FBBF6D101074EC14AFAC2D4F7\\
EFAC4CF9FEC1ED11BAE336D27D527665

The point C = z1*P + z2*Q has the coordinates:

x_C = 297009809158179528743712049839382569 9\\
042275210799431965163268798205921093339 5

x_C = 0x41AA28D2F1AB148280CD9ED56FED\\
A41974053554A42767B83AD043FD39DC0493

y[C] = 328425352786846634770946653225170845 0\\
680472103245454326813285455653927406091 0

y[C] = 0x489C375A9941A3049E33B34361DD\\
204172AD98C3E5916DE27695D22A61FAE46E

Then the parameter R = x_C (mod q) takes the value:

R = 297009809158179528743712049839382569 9\\
042275210799431965163268798205921093339 5

R = 0x41AA28D2F1AB148280CD9ED56FED\\
A41974053554A42767B83AD043FD39DC0493

Since the equality R = r holds, the digital signature is accepted.
```

8. Security Considerations

This entire document is about security considerations.

Current cryptographic resistance of GOST R 34.10-2001 digital signature algorithm is estimated as 2^128 operations of multiple elliptic curve point computations on prime modulus of order 2^256.

9. References

9.1. Normative References

[GOST3410] "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 Russia for Standards, 2001. (In Russian)

[GOST3411] "Information technology. Cryptographic Data Security. Hashing function.", GOST R 34.10-94, Gosudarstvennyi Standard of Russian Federation, Government Committee of Russia for Standards, 1994. (In Russian)

[RFC4357] Popov, V., Kurepkin, I., and S. Leontiev, "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", RFC 4357, January 2006.

9.2. Informative References

[ISO2382-2] ISO 2382-2 (1976), "Data processing - Vocabulary - Part 2: Arithmetic and logic operations".

[ISO9796-1991] ISO/IEC 9796:1991, "Information technology -- Security techniques -- Digital signature schemes giving message recovery."

[ISO14888-1] ISO/IEC 14888-1 (1998), "Information technology - Security techniques - Digital signatures with appendix - Part 1: General".

[ISO14888-2] ISO/IEC 14888-2 (1999), "Information technology - Security techniques - Digital signatures with appendix - Part 2: Identity-based mechanisms".

[ISO14888-3] ISO/IEC 14888-3 (1998), "Information technology - Security techniques - Digital signatures with appendix - Part 3: Certificate-based mechanisms".

[ISO10118-1] ISO/IEC 10118-1 (2000), "Information technology - Security techniques - Hash-functions - Part 1: General".

[ISO10118-2] ISO/IEC 10118-2 (2000), "Information technology - Security techniques - Hash-functions - Part 2: Hash-functions using an n-bit block cipher algorithm".

[ISO10118-3] ISO/IEC 10118-3 (2004), "Information technology - Security techniques - Hash-functions - Part 3: Dedicated hash-functions".

[ISO10118-4] ISO/IEC 10118-4 (1998), "Information technology - Security techniques - Hash-functions - Part 4: Hash-functions using modular arithmetic".

Appendix A. Extra Terms in the Digital Signature Area

The appendix gives extra international terms applied in the considered and allied areas.

1. Padding: Extending a data string with extra bits [ISO10118-1].

2. Identification data: A list of data elements, including specific object identifier, that belongs to the object and is used for its denotation [ISO14888-1].

3. Signature equation: An equation, defined by the digital signature function [ISO14888-1].

4. Verification function: A verification process function, defined by the verification key, which outputs a witness of the signature authenticity [ISO14888-1].

5. Signature function: A function within a signature generation process, defined by the signature key and by the digital signature scheme parameters. This function inputs a part of initial data and, probably, a pseudo-random number sequence generator (randomizer), and outputs the second part of the digital signature.

Appendix B. Contributors

Dmitry Kabelev
Cryptocom, Ltd.
14 Kedrova St., Bldg. 2
Moscow, 117218
Russian Federation

EMail: kdb@cryptocom.ru

Igor Ustinov
Cryptocom, Ltd.
14 Kedrova St., Bldg. 2
Moscow, 117218
Russian Federation

EMail: igus@cryptocom.ru

Sergey Vyshensky
Moscow State University
Leninskie gory, 1
Moscow, 119991
Russian Federation

EMail: svysh@pn.sinp.msu.ru

Author's Address

Vasily Dolmatov, Ed.
Cryptocom, Ltd.
14 Kedrova St., Bldg. 2
Moscow, 117218
Russian Federation

EMail: dol@cryptocom.ru