Blind Signature Protocols from Digital Signature Standards

Nikolay A. Moldovyan

St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences
14 Liniya, 39, St. Petersburg 199178, Russia (Email: nmold@mail.ru)

(Received Aug. 3, 2009; revised and accepted May 17 & July 12, 2010)

Abstract

Using Russian digital signature (DS) standards as the underlying scheme there are designed the blind DS protocols that are the first known implementation of the blind DS based on signature standards. There are also proposed blind collective DS protocols based on the DS standards. The last protocols are also the first implementation of the blind multi-signature schemes using the signature verification equations specified by DS standards.

Keywords: Blind collective signature, blind signature, collective digital signature, digital signature standard

1 Introduction

The digital signature (DS) protocols are widely used in information systems to solve different practical problems of the messages authentication. A variety of the DS protocols has been proposed in the literature [4, 9, 19], including multi-signature schemes [1, 7, 17]. A particular type of the protocols, called blind signature schemes [2], are especially interesting for application in the electronic money systems and in the electronic voting systems. The properties of the blind signatures are [17]:

1) The signer can not to read the document during process of signature generation;

2) The signer can not correlate the signed document with the act of signing.

The first property is provided by variety of DS algorithms in which the signature generation procedure uses the hash function value computed from the document to be signed. Actually, some user U is able to compute the hash value and to keep the document in secret. Then he can submit the document for signing and get the DS relating to the document. However the second property is not satisfied with this mechanism, since the signer can correlate the signature (if it is provided to him) with the act of signing. To make such correlation it is enough to keep records of every blind signature and hash function value submitted for signing.

The problem of providing the second property is known as anonymity (or untraceability) problem. To solve this problem there are used specially designed DS algorithms. There are known blind signature schemes based on difficulty of the factorization problem [3] and on difficulty of finding discrete logarithm [15]. Usually, the blind signature scheme is designed on the basis of some known DS algorithm, for example the RSA algorithm [16] or Schnorr's DS algorithm [15, 18].

To provide the anonymity of the signature and hash function value (or message submitted for signing) there are used so called *blinding factors*. Prior to submit a hash function value (or message M) for signing the user U computes the hash function value H and multiplies H (or M) by a random number (blinding factor). Then the user submit the blinded hash function value (or blinded document) for signing. The signer signs the blinded value H (or M) producing the blinded signature that is delivered to user U. The user divides out the blinding factor producing the valid signature to the original hash function value (or directly to the original document).

For practical applications it is interesting to use the blind signature schemes based on the DS algorithms specified by the DS standards. This paper is devoted to construction of the blind signature protocol based on Russian DS standards GOST R 34.10-94 and GOST R 34.10-2001.

In the second section there are proposed blind signature schemes based on these standards. The third section presents the implementation of the blind collective signature schemes [14] using the DS standards. The length of the blind collective DS does not depend on the number of the signers sharing the signature. Section 4 presents discussion on performance and security of the proposed protocols. It is shown that using the blind collective DS protocols requires performing the procedure on testing the public key correctness. There are formulated several reductionist security claims. An approach to give the arguments to the claims is proposed. The fifth section concludes the paper.

2 Blind Signature Protocols Based on Russian Standards

2.1 Using GOST R 34.10-94

The standard GOST R 34.10-94 [5] specifies the following signature verification equation

$$r = \left(g^{s/h} Y^{-r/h} \bmod p\right) \bmod q, \tag{1}$$

where p a prime such that $p-1$ contains a large prime factor q; the value g is generator of the q order subgroup in $\mathbb{F}_p^*$; Y is the public key computed as $Y = g^z \bmod p$; z is the secret key ($\mathbb{F}_p^*$ denotes the multiplicative group of the finite field $\mathbb{F}_p$). The signature generation to some message M is described as follows.

1) Generate a random value k and compute the value $\rho = g^k \bmod p$. Then compute the value $r = \rho \bmod q$ which is the first element of the signature.

2) Using the hash function F_h specified by the standard compute the hash value h from the message M.

3) Using the secret key compute the value

$$s = kh + zr \bmod q,$$

which is the second element of the signature.

Verification of the signature (r, s) to the message M is performed as follows:

1) Compute the hash value h from the message M: $h = F_h(M)$.

2) Compute the value $r^* = \left(g^{s/h} Y^{-r/h} \bmod p\right) \bmod q$.

3) Compare values r^* and r. If $r^* = r$, then the signature is valid. Otherwise the signature is rejected.

The described DS algorithms can be put into the base of some blind signature protocol using the blinding factors δ, τ, $Y^\mu \bmod p$, and $g^\epsilon \bmod p$, where the numbers $0 < \delta < q$, $0 < \tau < q$, $0 < \mu < q$, and $0 < \epsilon < q$ are selected at random. The blinded signature generation procedure is provided with the following blind signature protocol based on the standard GOST R 34.10-94. Two persons participate in the protocol, namely, the signer and the user U having intention to get a blind signature to the message M.

1) The signer generates the random value k, computes the value $\rho = g^k \bmod p$ and sends ρ to the user U.

2) The user U computes the hash value h' from the message M: $h' = F_h(M)$. Then he generates random values $\tau, \mu, \epsilon, \delta \in \{1, 2, \cdots, q-1\}$ and computes the blinded value $h = \tau h'$ and values

$$\begin{aligned} \rho' &= \rho^{1/\delta} Y^\mu g^\epsilon \bmod p, && (2)\\ r' &= \rho' \bmod q, \\ r &= \tau\delta(r' + \mu h') \bmod q. && (3) \end{aligned}$$

The value r' is the first element of the signature to message M.

3) The user U sends the values h and r to the signer.

4) The signer computes the value $s = kh + zr \bmod q$ and sends s to the user U.

5) The user U computes the second element s' of the signature to message M:

$$s' = \left(\tau^{-1}\delta^{-1}s + \epsilon h'\right) \bmod q. \tag{4}$$

The signature (r', s') is a valid signature to message M.

Correctness proof of the protocol. The element s of the blinded signature computed at step 4 satisfies the equation $s = kh + zr \bmod q$, therefore we have the congruences

$$\begin{aligned} g^s &\equiv g^{(kh+zr)} \equiv g^{kh} g^{zr} \bmod p \\ \Rightarrow \rho &\equiv g^k \equiv g^{s/h} g^{-zr/h} \bmod p. && (5) \end{aligned}$$

Taking into account that from (4) we have the equality $r' = \left(\tau^{-1}\delta^{-1}r - \mu h'\right) \bmod q$ the right part of the signature verification Equation (1) can be written as follows.

$$\begin{aligned} & \left(g^{\frac{s'}{h'}} Y^{-\frac{r'}{h'}} \bmod p\right) \bmod q \\ = & \left(g^{\frac{\tau^{-1}\delta^{-1}s + \epsilon h'}{h'}} Y^{-\frac{\tau^{-1}\delta^{-1}r - \mu h'}{h'}} \bmod p\right) \bmod q \\ = & \left(g^{\frac{s}{\delta\tau h'} + \epsilon} Y^{-\frac{r}{\delta\tau h'} + \mu} \bmod p\right) \bmod q \\ = & \left(g^{\frac{s}{\delta h}} g^\epsilon Y^{\frac{r}{\delta h}} Y^\mu \bmod p\right) \bmod q \\ = & \left(\left(g^{\frac{s}{h}} Y^{-\frac{r}{h}}\right)^{1/\delta} g^\epsilon Y^\mu \bmod p\right) \bmod q \\ = & \left(\rho^{1/\delta} Y^\mu g^\epsilon \bmod p\right) \bmod q \\ = & \rho' \bmod q \\ = & r'. \end{aligned}$$

The right part of the signature verification equation is equal to the signature element r', therefore the signature is valid. Thus, the protocol performs correctly. The produced signature (r', s') is known for user U and unknown for the signer.

The protocol provides anonymity of the user in the case when the message M and signature (r', s') will be disclosed to the signer. The disclosed signature and document can be correlated with each tetrad (ρ, r, s, h) recorded by the signer (it is supposed the signer records in a file all tetrads (ρ, r, s, h) produced by each of the performed blind DS procedures), since there exists a quadruple of the values $\tau, \mu, \epsilon, \delta \in \{1, 2, \cdots, q-1\}$ such that Equations (2), (3), and (4) hold, for each of the tetrads (ρ, r, s, h).

Indeed, it can be shown that with probability $1 - q^{-1} \approx 1$ for arbitrary of the mentioned correlations there exists a unique quadruple $(\tau, \mu, \epsilon, \delta)$ satisfying Equations (2)-(4), therefore all of the correlations have the same probability. Actually, the value τ is defined by formula $\tau =$

$H/H' \bmod q$. The values μ, ϵ, δ can be computed from Equations (2)-(4) as follows. Note that $\delta^{-1}k + z\mu + \epsilon = \log_g \rho' = L$, where $L < q$, since accordingly to the blind signature protocol the value ρ' is computed as an integer power of g (see Formula (2)): $\rho' = g^{k/\delta}g^{z\mu}g^{\epsilon}$. The value ρ' is computed as an intermediate value, while performing the signature verification procedure. Thus, taking into account the relation between r and r', s and s', we get the following system of three linear congruences with unknowns $\mu, \epsilon, \delta^{-1}$:

$$\left\{\begin{array}{rcl} \delta^{-1}k + z\mu + \epsilon & \equiv & L \bmod q \\ \tau^{-1}\delta^{-1}r - \mu h' & \equiv & r' \bmod q \\ \tau^{-1}\delta^{-1}s + \epsilon h' & \equiv & s' \bmod q. \end{array}\right.$$

With very low probability, equal to $q^{-1} < 2^{160}$ the determinant of this system is equal to zero, therefore practically in all cases this system has solution. This means that arbitrary disclosed signature r', s' can be associated with arbitrary tetrad (ρ, r, s, h) recorded by the signer with the unique quadruple $(\tau, \mu, \epsilon, \delta)$, where δ is computed as $\delta = \left(\delta^{-1}\right)^{-1} \bmod q$.

2.2 Using GOST R 34.10-2001

The standard GOST R 34.10-2001 [6] specifies a DS algorithm based on elliptic curves (ECs) over finite field (for details of the application of the ECs in cryptography see [10, 12]). The specified EC is described by the following equation

$$y^2 = x^3 + ax^2 + b \bmod p, \tag{6}$$

where p is a prime and coefficients a and b are selected so that the EC order contains a large prime factor q. Points of the EC are pairs of numbers x and y ($0 < x < p$, $0 < y < p$) called abscissa and ordinate, which satisfy Equation (6). The EC represents a commutative finite group with the point addition operation as the group operation. The multiplication of some EC point A by number m is defined as $mA = A + A + \cdots + A$ (m times). The neutral element of the group of the EC points is the point in infinity denoted O. On definition we have $A + O = O + A = A$ and $mO = O$.

The addition of the points $A = (x_A, y_A)$ and $B = (x_B, y_B)$ is performed with the following formulas for computing the abscissa x_C and ordinate y_C of the point $C = A + B$:

$$\begin{array}{rcl} x_C & = & \lambda^2 - x_A - x_B \bmod p \\ y_C & = & \lambda(x_A - x_C) - y_A \bmod p, \end{array}$$

where

$$\lambda = \left\{\begin{array}{ll} \frac{y_B - y_A}{x_B - x_A} \bmod p, & \text{if } A \neq B \\ \frac{3x_A^2 + a}{2y_A} \bmod p, & \text{if } A = B. \end{array}\right.$$

Subtraction of the points B and $A = (x_A, y_A)$ is defined as follows $B - A = B + (-A)$, where $-A = (x_A, -y_A)$.

In GOST R 34.10-2001 the public key is some EC point Q computed as follows $Q = zG \bmod p$, where z is the secret key and G is the EC point having the order q. The signature to some message M is generated as follows:

1) Generate a random value k, compute the point $C = kG$ and define $r = x_C$. The value r is the first element of the signature.

2) Using the hash function F_h specified by the standard compute the hash value h from the message M: $h = F_h(M)$. Then it is computed value $e = h \bmod q$.

3) Using the secret key compute the value $s = ke + zr \bmod q$, which is the second element of the signature.

Verification of the signature (r, s) to the message M is performed as follows:

1) Compute the hash value h from the message M: $h = F_h(M)$. Then compute $e = h \bmod q$.

2) Compute the point $C^* = \left(e^{-1}s \bmod q\right)G - \left(e^{-1}r \bmod q\right)Q$. Define $r^* = x_{C^*}$, where x_{C^*} is the abscissa of the point C^*.

3) Compare values r^* and r. If $r^* = r$, then the signature is valid. Otherwise the signature is rejected.

Using the described DS algorithms one can compose a blind signature protocol like in the case of GOST R 34.10-94. In the case of the blind DS protocol based on the standard GOST R 34.10-2001 there are also used the blinding parameters $\delta, \tau, \mu, \epsilon \in \{1, 2, \cdots, q-1\}$, which are generated at random. The blind signature protocol based on the standard GOST R 34.10-2001 is described as follows.

1) The signer generates the random value k, computes the point $C = kG$ and defines $r = x_C$. The value r is sent to the user U.

2) The user U computes the hash value h' from the message M: $h' = F_h(M)$ and then the value $e' = h' \bmod q$. Then he generates random values $\tau, \mu, \epsilon, \delta \in \{1, 2, \cdots, q-1\}$ and computes the blinded value $e = \tau e'$, the point $C' = (\delta^{-1} \bmod q)C + \mu Q + \epsilon G$, $r' = x_{C'}$, and $r = \tau\delta(r' + \mu h') \bmod q$ (r' is the first element of the DS to message M.)

3) The user U sends the value r to the signer.

4) The signer computes the value $s = ke + zr \bmod q$ and sends s to the user U.

5) The user U computes the second element s' of the signature to message M:

$$s' = \left(\tau^{-1}\delta^{-1}s + \epsilon h'\right) \bmod q.$$

The signature (r', s') is a valid signature to message M.

Correctness proof of the protocol. The element s of the blinded signature computed at step 4 satisfies the equation $s = ke + zr \bmod q$, therefore we have the equality

$$\begin{aligned} sG &= (ke + zr \bmod q)G = keG + zrG \\ \Rightarrow C &= kG = (se^{-1} \bmod q)G - (zre^{-1} \bmod q)G. \end{aligned}$$

Taking into account that $r' = \left(\tau^{-1}\delta^{-1}r - \mu e'\right) \bmod q$ one can write

$$\begin{aligned} C^* &= \left(\frac{s'}{e'} \bmod q\right) G - \left(\frac{r'}{e'} \bmod q\right) Q \\ &= \left(\frac{\tau^{-1}\delta^{-1}s + \epsilon e'}{e'} \bmod q\right) G \\ &\qquad - \left(-\frac{\tau^{-1}\delta^{-1}r - \mu e'}{e'} \bmod q\right) Q \\ &= \left(\frac{1}{\delta} \bmod q\right) \left(\left(\frac{s}{e} \bmod q\right) G - \left(\frac{r}{e} \bmod q\right) Q\right) \\ &\qquad + \epsilon G + \mu Q \\ &= \left(\frac{1}{\delta} \bmod q\right) C + \epsilon G + \mu Q \\ &= C' \\ \Rightarrow r^* &= x_{C^*} = x_{C'} = r'. \end{aligned}$$

Thus, the protocol performs correctly. The produced signature (r', s') is known for user U and unknown for the signer. The protocol provides anonymity of the user in the case when the message M and signature (r', s') will be disclosed to the signer. With the same probability the disclosed signature and document can be associated to each tetrad (C, r, s, h) recorded by the signer. This fact can be demonstrated like in the case of the blind DS protocol based on the GOST R 34.10-94.

3 Blind Collective Signature Schemes from Russian DS Standards

3.1 Schemes Based on GOST R 34.10-94

To implement the blind collective DS protocol based on standard GOST R 34.10-94 we have used the design of the collective DS based on this standard, which was proposed earlier in [13]. In that construction we have introduced the blinding mechanism described above and get the following protocol.

The user U and m signers participate in the protocol. The public keys of the signers are $Y_i = g^{z_i} \bmod p$, where $i = 1, 2, \cdots, m$ and z_i is the secret key of the ith signer. The user U has intention to get a blind signature to some message M.

1) Each ith signer generates the random value k_i, computes the value $\rho_i = g^{k_i} \bmod p$, where $i = 1, 2, \cdots, m$. Then it is computed the common value $\rho = \prod_{i=1}^{m} \rho_i \bmod p = g^{\sum_{i=1}^{m} k_i \bmod q} \bmod p$.

2) The value ρ is sent to the user U.

3) The user U computes the hash value h' from the message M: $h' = F_h(M)$. Then he generates random values $\tau, \mu, \epsilon, \delta \in \{1, 2, \cdots, q-1\}$ and computes the blinded values $h = \tau h'$, $\rho' = \rho^{1/\delta} Y^{\mu} g^{\epsilon} \bmod p$, $r' = \rho' \bmod q$, and $r = \tau\delta(r' + \mu h') \bmod q$. (The value r' is the first element of the signature to message M.)

4) The user U sends the values r and h to the signer.

5) Each ith signer computes the value $s_i = k_i h + z_i r \bmod q$, where $i = 1, 2, \cdots, m$. Then the signers compute the common value $s = \sum_{i=1}^{m} s_i \bmod q$.

6) The value s is sent to the user U.

7) The user U computes the second element s' of the collective signature to message M:

$$s' = \left(\tau^{-1}\delta^{-1}s + \epsilon h'\right) \bmod q.$$

The signature (r', s') is a valid signature to message M. The collective signature verification is performed with the verification Equation (1), where Y is the collective public key computed as product of individual public keys of all signers, i.e. $Y = \prod_{i=1}^{m} Y_i \bmod p$.

Correctness proof of the protocol. Each share s_i of the second element s of the blind collective signature, which is computed at step 4 satisfies the equation $s_i = k_i h + z_i r \bmod q$, therefore we have the congruences

$$\begin{aligned} g^{s_i} &\equiv g^{(k_i h + z_i r)} \equiv g^{k_i h} g^{z_i r} \bmod p \\ g^{\sum_{i=1}^{m} s_i} &\equiv g^{h \sum_{i=1}^{m} k_i} gr \sum_{i=1}^{m} z_i \bmod p \\ \rho &= g^{\sum_{i=1}^{m} k_i} \bmod p = g^{s/h} Y^{-r/h} \bmod p, \end{aligned}$$

where $Y = \prod_{i=1}^{m} Y_i \bmod p = g^{\sum_{i=1}^{m} z_i} \bmod p$. Taking into account the equality $r' = \left(\tau^{-1}\delta^{-1}r - \mu h'\right) \bmod q$ the right part of the collective signature verification equation can be written as follows

$$\begin{aligned} & \left(g^{\frac{s'}{h'}} Y^{-\frac{r'}{h'}} \bmod p\right) \bmod q \\ = & \left(g^{\frac{\tau^{-1}\delta^{-1}s + \epsilon h'}{h'}} Y^{-\frac{\tau^{-1}\delta^{-1}r - \mu h'}{h'}} \bmod p\right) \bmod q \\ = & \left(g^{\frac{s}{\delta\tau h'} + \epsilon} Y^{-\frac{r}{\delta\tau h'} + \mu} \bmod p\right) \bmod q \\ = & \left(g^{\frac{s}{\delta h}} g^{\epsilon} Y^{\frac{r}{\delta h}} Y^{\mu} \bmod p\right) \bmod q \\ = & \left(\left(g^{\frac{s}{h}} Y^{-\frac{r}{h}}\right)^{1/\delta} g^{\epsilon} Y^{\mu} \bmod p\right) \bmod q \\ = & \left(\rho^{1/\delta} Y^{\mu} g^{\epsilon} \bmod p\right) \bmod q \\ = & \ \rho' \bmod q \\ = & \ r'. \end{aligned}$$

The right part of the signature verification equation is equal to the signature element r', therefore the signature is valid. Thus, the protocol performs correctly. The produced collective signature (r', s') is known for user U and unknown for the signers.

The protocol provides anonymity of the user in the case when the message M and signature (r', s') will be disclosed to the signer. With the same probability the disclosed signature (r', s') and document M can be correlated with each tetrad (ρ, r, s, h) recorded by the signers (it is supposed the signers record all tetrads (ρ, r, s, h) produced while performing the blind DS procedures). This property of the protocol can be demonstrated like in the case of the blind DS protocol described in Section 2.1.

3.2 Scheme Based on GOST R 34.10-2001

Using GOST R 34.10 2001 the collective DS generation is designed in the following way. Suppose m signers possessing public keys $Q_i = z_i G$, where z_i is the secret key of the ith signer $(i = 1, 2, \cdots, m)$ are to sign a document M with a single signature. This task is solved by the following protocol.

1) Each ith signer selects at random a value k_i and computes the EC point $C_i = k_i G$, where G is the q order point of the EC (q is a prime).

2) It is computed the common randomization point $C = C_1 + C_2 + \cdots + C_m$ and the randomization value $r = x_C \bmod q$. The value r is the first part of the collective DS.

3) It is computed the hash value from the document $h = F_H(M)$ and the value $e = h \bmod q$.

4) Each user computes his share in the composite DS as follows $s_i = (z_i r + k_i e) \bmod q$, where $d_i < q$ is the secret key of the ith user, $e = H \bmod q$, H Is the hash function value.

5) The second part of the collective signature is $s = \sum_{i=1}^{m} s_i \bmod q$. The full collective DS is (r, s).

The signature (r, s) is a valid collective signature to message M. The verification of the signature (r, s) is performed as follows.

1) Compute the collective public key as the point $Q = \sum_{i=1}^{m} Q_i$.

2) Compute the EC point $C^* = \left(se^{-1} \bmod q\right) G - \left(re^{-1} \bmod q\right) Q$.

3) Compute the value $r^* = x_{C^*} \bmod q$ and compare r^* and r. If $r^* = r$, then the collective DS is valid.

Combining this protocol with the blind signature protocol based on GOST R 34.10-2001 we have constructed the following blind collective signature protocol.

1) Each ith signer selects at random a value k_i and computes the EC point $C_i = k_i G$. Then they compute the common randomization point $C = C_1 + C_2 + \cdots + C_m$.

2) The point C is sent to the user U that is going to get a blind signature to some message M.

3) The user U computes the hash value h' from the message M: $h' = F_h(M)$ and then the value $e' = h' \bmod q$. Then he generates random values $\tau, \mu, \epsilon, \delta \in \{1, 2, \cdots, q-1\}$ and computes the value $e = \tau e'$, the point $C' = \left(\delta^{-1} \bmod q\right) C + \mu Q + \epsilon G$, $r' = x_{C'}$, and $r = \tau\delta(r' + \mu h') \bmod q$ (r' is the first element of the DS to message M.)

4) The user U sends the value r to the signer.

5) The ith signer computes the value $s_i = k_i e + z_i r \bmod q$, $i = 1, 2, \cdots, m$.

6) The signers compute the second element s of the collective blind signature as follows $s = \sum_{i=1}^{m} s_i \bmod q$. The blind collective DS is (r, s).

7) The value (r, s) is sent to user U.

8) The user U computes the second element s' of the collective signature to message M:

$$s' = \left(\tau^{-1}\delta^{-1}s + \epsilon h'\right) \bmod q.$$

The signature (r', s') is a valid collective signature to message M.

Correctness proof of the blind collective DS protocol. Each share s_i of the second element of the blinded collective signature s computed at step 4 satisfies the equation $s_i = k_i e + z_i r \bmod q$, therefore we have the equality

$$\begin{aligned}
s_i G &= (k_i e + z_i r \bmod q)G = k_i e G + z_i r G \\
C_i &= k_i G = (s_i e^{-1} \bmod q)G - (z_i r e^{-1} \bmod q)G \\
C &= \sum_{i=1}^{m} C_i = \left(\frac{1}{e}\sum_{i=1}^{m} s_i \bmod q\right) G \\
&\qquad - \left(\frac{r}{e}\sum_{i=1}^{m} z_i \bmod q\right) G \\
&= \left(se^{-1} \bmod q\right) G - \left(re^{-1} \bmod q\right) Q.
\end{aligned}$$

Taking into account the equality $r' = \left(\tau^{-1}\delta^{-1}r - \mu e'\right) \bmod q$ the right part of the signature verification equation can be written as follows.

$$\begin{aligned}
&\left(\frac{s'}{e'} \bmod q\right) G - \left(\frac{r'}{e'} \bmod q\right) Q \\
= &\left(\frac{\tau^{-1}\delta^{-1}s + \epsilon e'}{e'} \bmod q\right) G \\
&- \left(\frac{\tau^{-1}\delta^{-1}r - \mu e'}{e'} \bmod q\right) Q \\
= &\left(\frac{1}{\delta} \bmod q\right) \left(\left(\frac{s}{e} \bmod q\right) G - \left(\frac{r}{e} \bmod q\right) Q\right) \\
&+\epsilon G + \mu Q \\
= &\left(\frac{1}{\delta} \bmod q\right) C + \epsilon G + \mu Q = C' \\
r^* = & x_{C^*} = x_{C'} = r'.
\end{aligned}$$

The right part of the signature verification equation is equal to the signature element r', therefore the signature is valid. Thus, the protocol performs correctly. The produced signature (r', s') is known for the user U and unknown for the signers. The protocol provides anonymity of the user in the case when the message M and signature (r', s') will be disclosed to the signer. The disclosed signature and document can be associated to each tetrad (C, r, s, h) recorded by the signer. This fact can be demonstrated like in the case of blind signature considered in Section 2.1.

4 Discussion on Performance and Security

Performance of the proposed crypto-schemes is defined mainly by the signature generation and verification procedures specified by the used DS standards. To get higher performance of the blind signature, collective signature, and blind collective signature protocols it is efficient to use the DS standards providing higher performance, for example the USA standards DSA and ECDSA [8]. Unfortunately, majority of the official standards including DSA and ECDSA do not allow developing such protocols without modifying their specified signature generation and verification procedures. Trying different other DS standards we have succeeded to design the protocols based on the Russian signature standards GOST R 34.10-94 and GOST R 34.10-2001 possessing sufficiently high performance for variety of different practical applications.

The performance of the protocols can be roughly estimated taking into account only the exponentiation and EC point multiplication operations. The signature verification in the designed protocols takes two exponentiation operations in the case of using GOST R 34.10-94 or two EC point multiplication operations in the case of GOST R 34.10-2001. The blind signature generation takes 4 operations. The collective DS generation takes m operations in the case of the DS shared by m signers. The blind collective DS generation takes $m + 3$ operations in the case of the DS shared by m signers. Table 1 presents comparison of the computation complexity of the protocols and the Russian DS standards in the case of 80-bit security. (Note that the Russian standards specify the minimum security level equal to 80-bit security for GOST R 34.10-94 and to 128-bit security for GOST R 34.10-2001. Indeed, the GOST R 34.10-94 specifies using 1024-bit modulus and the GOST R 34.10-2001 specifies using the minimum size of the ground field characteristic equal to 256 bits. However to compare the performance it is reasonable to consider the same security level for the both standards.)

The GOST R 34.10-2001 possesses higher performance than GOST R 34.10-94. Indeed, the EC point multiplication operation takes about 2400 multiplications modulo 160-bit prime against 240 exponentiation operations modulo 1024-bit prime, which have about the same difficulty as 9600 multiplications modulo 160-bit prime. Therefore in the case of 80-bit security the GOST R 34.10-2001 is about 4 times faster than GOST R 34.10-94.

The designed blind, collective, and blind collective signature protocols are based on the standards GOST R 34.10-94 and GOST R 34.10-2001, therefore the security of the protocol depends on the security of the standards that relates to the DS schemes based on difficulty of finding discrete logarithm. In accordance with [11, 15] among this type of digital signatures there are DS schemes with provable security. An example of such DS schemes is the Schnorr's signature algorithm [18]. The formal proof of the security of such DS schemes uses the possibility to force the forgery program (for details see [11, 15]) to use the same value of the signature randomization parameter $\rho = g^k \bmod p$ to produce two different signatures. This possibility is connected with the computing the hash function value h from the message to which the parameter ρ is concatenated: $h = F_h(M, \rho)$. This design feature require generation of the value ρ before computing the hash function. Therefore it appears possibility to change suddenly the hash function and get two different hash values computed using the same value $\rho : h = F_h(M, \rho)$ and $h' = F'_h(M, \rho) \neq h$. (In the formal security proof it is supposed that two copies of the forgery program are executed on to different computers using the same sequence of random bits that are used to make choices at various points in the work of the programs).

However, like in the USA standards DSA and ECDSA [8] in the standards GOST R 34.10-94 and GOST R 34.10-2001 the hash fuction is evaluated only as a function of the massage M, i.e. $h = F_h(M)$, and the value h does not depend on the randomly generated parameter ρ. Therefore the reductionist security argument in line with [11, 15] is not possible, since the forgery program cannot be forced to use the same value ρ while producing two different signatures for M with different values h. At present there is no known argument that shows the equivalence of the ability to forge GOST R 34.10-94 or GOST R 34.10-2001 with the discrete logarithm problem.

Nevertheless the GOST R 34.10-94 or GOST R 34.10-2001 (like DSA and ECDSA) are official standards that

Table 1: Comparison of the computation complexity (in multiplications modulo 160-bit prime) of the proposed protocols and DS standards in the case of the 80-bit security.

DS scheme	GOST R 34.10-94		GOST R 34.10-2001	
	DS generation	DS verification	DS generation	DS verification
standard	2400	4800	600	1200
blind	9600	4800	2400	1200
collective	$2400m$	4800	$600m$	1200
blind collective	$2400m + 7200$	4800	$600m + 1800$	1200

have sufficiently wide practical application, their security is based only on detailed security examination by the top experts though.

In the case of collective DS protocol there is used modified verification equation. The modification consists in using the collective public key instead of individual one. This is a source of the following specific attack that is possible, if the certification authority does not perform the correctness verification of the public keys. If an attacker gets a certificate containing his public key Y' computed as $Y' = g^z Y_1^{-1} Y_2^{-1}$, where Y_1 and Y_2 are the public keys of some users, then for arbitrary messages he will be able to generate the collective DS corresponding to the collective public key $Y_{\text{coll}} = Y' Y_1 Y_2 = g^z$. Indeed the collective DS corresponding to this collective public key is generated like individual DS, using the value z. This attack can be easily extended to arbitrary number of signers. Such attacks based on incorrect generation of public keys are possible, if there are used no public key correctness verification procedures. Thus, in the collective DS protocol we strongly need to provide correctness of the public key structure. This problem has simple and natural solution that consists in the following. Before to issue a digital certificate notifying a public key of some user the certification authority has to request the user to sign some message. If public key is correctly generated, then the user will be able to generate a valid signature. If the user does not generate such test signature, then he is considered as potential attacker.

Let us consider security of the proposed collective DS protocol in the case of using public keys correctness of which is approved. Suppose it is given a set of public keys $\{Y_1, Y_2, \cdots, Y_m, \cdots\}$ authenticated by the certification authority. To forge a collective signature means computing a signature (r^*, s^*) satisfying the verification equation written for some collective public key.

Claim 1. *Any successful attack breaking the collective DS protocol based on the DS standard GOST R 34.10-94 also breaks the GOST R 34.10-94.*

Claim 2. *Any successful attack breaking the collective DS protocol based on the GOST R 34.10-2001 also breaks the DS standard GOST R 34.10-2001.*

Argument. The participants of the collective DS protocol have significant more possibilities to attack the protocol than outsiders. Therefore it is reasonable to consider the following message forgery attack against the collective DS protocol based on the standard GOST R 34.10-94 (consideration of the collective DS protocol based on GOST R 34.10-2001 is analogous).

Suppose it is given a message M and $m-1$ signers attempts to create a collective DS corresponding to m signers owning the collective public key $Y = Y^* Y_m \bmod p$, where $Y^* = \prod_{i=1}^{m-1} Y_i \bmod p$, i.e. $m-1$ users unite their efforts to generate a pair of numbers (r^*, s^*) such that equation

$$r^* = \left(Y^{-r^*/h} g^{s^*/h} \bmod p\right) \bmod q$$

holds. Suppose that they are able to do this, i.e. the collective forger (i.e. the considered $m-1$ signers) is able to calculate a valid signature (r^*, s^*) corresponding to collective public key $Y = Y_1 Y_2 \cdots Y_m \bmod p$. The collective DS satisfies the following equations:

$$\begin{aligned} r^* &= \left(Y^{-r^*/h} g^{s^*/h} \bmod p\right) \bmod q \\ &= \left(Y_m^{-r^*/h} Y^{*-r^*/h} g^{s^*/h} \bmod p\right) \bmod q \\ &= \left(Y_m^{-r^*/h} g^{\frac{-r^* \sum_{i=1}^{m-1} z_i}{h}} g^{s^*/h} \bmod p\right) \bmod q \\ r^* &= \left(Y_m^{-r^*/h} g^{\frac{s^* - r^* \sum_{i=1}^{m-1} z_i}{h}} \bmod p\right) \bmod q. \end{aligned}$$

The last expression represents the signature verification equation specified by GOST R 34.10-94, which is written for the individual signature (r^*, s^{**}) of the mth user, where $s^{**} = \left(s^* - r^* \sum_{i=1}^{m-1} z_i\right) \bmod q$. Thus, the pair of numbers (r^*, s^{**}) is a forged signature of the mth user to message M, i.e. an attack breaking the collective DS scheme also breakes the Russian DS standard. □

Regarding to the blind collective protocols based on the Russian signature standards there are hold the following reductionist security claims.

Claim 3. *Any successful attack breaking the blind collective DS protocol based on the standard GOST R 34.10-94 also breaks the blind signature scheme based on the GOST R 34.10-94.*

Claim 4. *Any successful attack breaking the blind collective DS protocol based on the standard GOST R 34.10-2001 also breaks the blind signature scheme based on the*

GOST R 34.10-2001.

Using the analogy with the Claim 1 one can easily give the argument to each of the last two reductionist security claims that show the blind collective protocol is as secure as the underlying blind signature scheme is secure.

A secure blind signature scheme satisfies both the blindness and the non-forgebility properties. The blindness property of the proposed blind signature and blind colective signature protocols has been considered in Sections 2 and 3. Elaboration of the reductionist security argument for the non-forgebility property of these protocols is an open problem approaches to which are not evident. The argument technique [15] proposed for blind signature schemes based on discrete logarithm problem uses essentially peculiarity of computing the hash function value as $h = F_h(M, \rho)$, however the Russian DS standards are free of such peculiarity.

Some informal illustrative justification of the reductionist security claim that blind signature based on the GOST standards is as secure as these standards are secure against forgery attacks is the following one. Suppose there is known some attack on the blind signature, which provides possibility to compute $k+1$ signatures from some random k blind signatures. Then from $2k$ usual DS it is possible to compute an additional signature. Indeed, from Equation (6) written for some blind signature (r, s) and some blinded hash function value h we have

$$r = \left(g^k \bmod q\right) = \left(g^{s/h} Y^{-r/h} \bmod q\right) \bmod p,$$

i.e. the blind signature satisfies the verification equation therefore arbitrary two usual signatures (r, s) and (r', s') can be considered as a pair of the blind signature (r, s) and the signature (r', s') computed from the blind one (this can be easily proved like proving anonimity of the blind signature in Section 2.1). Thus, one can consider half of the $2k$ usual DS as k blind signatures and apply the supposed attack to generate an additional DS. The last means that the underlying DS scheme does not provide non-forgebility property, i.e. it is not secure. However we believe that the standards GOST R 34.10-94 and GOST R 34.10-2001 are secure DS algorithms.

5 Conclusion

Two novel items have been presented in the paper. For the first time the blind signature schemes have been implemented using the official DS standards as the underlying algorithm. New multi-signature schemes called blind collective DS protocols have been constructed on the base of the Russian DS standards GOST R 34.10-94 and GOST R 34.10-2001.

It is interesting to implement the mentioned DS schemes using some other official DS standards [8], first of all using the USA standards DSA and ECDSA. Our attempts to use the USA signature standards as the underlying algorithms in the blind signature protocols were not successful. Probably new ideas should be applied to design blind signature schemes based on DSA and ECDSA. Authors invite readers to try some approaches to this problem.

It seems that the blind collective DS protocols are promising for application in the electronic money systems in which the electronic banknotes are issued by several banks. Using DS standards as underlying signature schemes of the blind DS protocols appears to be attractive for practical applications.

Acknowledgments

The work supported by Russian Foundation for Basic Research grant # 10-07-90403-Ukr_a.

References

[1] A. Boldyreva, “Efficient threshold signature, multisignature and blind signature shemes based on the gap-Diffie-Hellman-group signature sheme”, LNCS 2139, Springer-Verlag, pp. 31-46, 2003.

[2] D. Chaum, “Blind signature systems”, U. S. Patent # 4-759-063, 19 July 1988.

[3] D. Chaum, “Security without identification: Transaction systems to make big brother obsolete”, *Communications of the ACM*, vol. 28, no. 10, pp. 1030-1044, 1985.

[4] S. S. M. Chow, “Multi-designated verifiers signatures revisited”, *International Journal of Network Security*, vol. 7. no. 3, pp. 348-357, 2008.

[5] Government Committee of the Russia for Standards, *Information Technology - Cryptographic Data Security - Produce and check procedures of Electronic Digital Signature based on Asymmetric Cryptographic Algorithm*, Russian Federation Standard: GOST R 34.10-94, 1994 (in Russian).

[6] Government Committee of the Russia for Standards, *Information Technology - Cryptographic Data Security - Produce and check procedures of Electronic Digital Signature*, Russian Federation Standard: GOST R 34.10-2001, 2001 (in Russian).

[7] M. S. Hwang and C. C. Lee, “Research issues and challenges for multiple digital signatures”, *International Journal of Network Security*, vol. 1, no. 1, pp. 1-7, 2005.

[8] ISO, *Information Technology - Security Techniques - Digital Signatures with Appendix - Part 3: Discrete Logarithm Based Mechanisms*, International Standard ISO/IEC 14888-3: 2006(E).

[9] R. S. Katti and R. G. Kavasseri, “Nonce generation for the digital signature standard”, *International Journal of Network Security*, vol. 11, no. 1, pp. 23-32, 2010.

[10] N. Koblitz, *Elliptic Curve Cryptosystems*, Mathematics of Computation Advances, vol. 48, pp. 203-209, 1987.

[11] N. Koblitz and A. J. Menezes, "Another Look at Provable Security", *Journal of Cryptology*, vol. 20, pp. 3-38, 2007.

[12] V. Miller, "Use of elliptic curves in cryptography", *Advances in cryptology: Proceedings of Crypto'85*, LNCS 218, pp. 417-426, 1986.

[13] N. H. Minh, N. A. Moldovyan, and N. L. Minh, "New multisignature protocols based on randomized signature algorithms", *2008 IEEE International Conference on Research, Innovation and Vision for the Future in computing & Communication Technologies*, PP. 23, Ho Chi Minh City, Vietnam, July 13-17, 2008.

[14] N. A. Moldovyan and A. A. Moldovyan, "Blind collective signature protocol based on discrete logarithm problem", *International Journal of Network Security*, vol. 11, no. 2, pp. 106-113, 2010.

[15] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures", *Journal of Cryptology*, vol. 13, pp. 361-396, 2000.

[16] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public key cryptosystems", *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[17] B. Schneier, "Applied Cryptography (2nd Ed.)", John Wiley & Sons, 1996.

[18] C. P. Schnorr, "Efficient signature generation by smart cards", *Journal of Cryptology*, vol. 4, pp. 161-174, 1991.

[19] Z. M. Zhao, "ID-based weak blind signature from bilinear pairings", *International Journal of Network Security*, vol. 7, no. 2, pp. 265-268, 2008.

Nikolay A. Moldovyan is an honored inventor of Russian Federation (2002), a laboratory head at St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences, and a Professor with the St. Petersburg State Electrotechnical University. His research interests include information security and cryptology. He has authored or co-authored more than 70 inventions and 230 scientific articles, books, and reports. He received his Ph.D. from the Academy of Sciences of Moldova (1981).