Protocols for Simultaneous Signing Contracts

Nguyen H. Minh
Le Qui Don Technical University
Ha Noi, Viet Nam
Email: minhnh@mail.ru

Nikolay A. Moldovyan
St. Petersburg Institute for Informatics and Automation of
Russian Academy of Sciences
St. Petersburg, Russia

***Abstract* – Signing electronic messages authentication is an issue of significant importance for computer and telecommunication systems. To solve efficiently some special practical problems the multisignature protocols are applied. New multisignature protocols based on the elliptic curves and discrete logarithm problem in finite groups are proposed to solve the problem of simultaneous signing contracts. The protocols provide simultaneous generation of the collective digital signature. There are considered the following two cases: 1) a group of signers are signing simultaneously one document and 2) several different groups of signers are signing several different documents. The signature verification is performed using collective public key computed from the whole set of individual public keys as the some convolution of this set.**

Keywords – Digital signature, collective digital signature, discrete logarithm problem, multisignature schemes, public key, simultaneous signing contracts, composite digital signature.

I. Introduction

Information authentication with digital signatures (DS) is widely used in large information system. Different practical problems connected with electronic information authentication are solved with different types of the DS protocols. There are developed the following protocols: group signature schemes, multisignature schemes, aggregate signatures, blind signatures [1-3].

There are known different constructions of the multisignature schemes [1, 2]. However only few of them provide the property of internal integrity that means no manipulation with the signature is possible during and after the signature generation. Recently a special type of the multisignature schemes called collective DS has been proposed [4]. The internal integrity property is provided by that approach. It can be applied to develop multisignature protocols based on difficulty of finding discrete logarithms in different types of finite groups. The collective DS protocol solves naturally the problem of simultaneous signing an electronic document (contract).

In present paper new variants of the collective DS protocols are developed and proposed for solving the problem of simultaneous signing a packet of electronic messages by different sets of signers.

The rest of the paper is organized as follows. In the second section the general structure of the collective DS protocols based on computing convolutions from some sets of individual parameters produced by signers is considered and used to design the collective DS based on Russian standard GOST R 34.10-2001. In the third section a new variant of the collective DS protocol, called composite DS, is proposed. The protocol allows one to produce a short signature confirming the fact that several different documents are signed by different sets of signers. The composite DS protocols suite well for simultaneous signing a package of contracts. It uses a special procedure for computing the collective public key. It is also proposed a method for transforming the collective DS protocol into the composite one. Section 4 concludes the paper.

II. Collective Signature Protocols

A. Collective Digital Signature

The collective DS protocol proposed in [4] is based on difficulty of finding the kth roots modulo prime $p = Nk^2 + 1$, where k is a large prime ($|k| \geq 160$ bit) and N is such even number that $|p| \geq 1024$ bit. That protocol uses the collective process of the formation of some common random parameter E that depends on the message to be signed and on some set of random values generated by each of the signers participating in the protocol. The parameter E is the first part of the DS, which is used individually by each signer to compute his share in the collective DS. Then the convolution S of all shares of the signers is computed as the second part of the collective DS (E, S). Generalized structure of this protocol is presented in Fig.1 and 2.

The protocol works as follows:

The individual public key is computed as the kth power of the private key X: $Y = X^k \bmod p$.

Suppose some subset of m users is to sign a message M with some single collective DS and the jth user owns the private key $X_j < p$ and the public key $Y_j = X_j^k \bmod p$, where $j = 1, 2,\ldots, m$.

The Collective DS algorithm defines the following signature generation procedure (see Fig.1):

1. Each user generates a random value $t_j < p$ and computes the value $R_j = t_j^k \bmod p$, where $j = 1, 2,\ldots, m$.

2. The common randomization value R is computed as a convolution of all individual values R_j:

The work supported by Russian Foundation for Basic Research grant # 08-07-00096-a.

978-1-4244-5139-5/09/$26.00 ©2009 IEEE

$$R = R_1R_2\ldots R_m \bmod p.$$

3. The first part E of the collective DS (E, S) is computed using some specified hash function F_H: $E = F_H(M, R)$.

4. Using the convolution R and individual values t_j each of the users computes its share in the collective DS:

$$S_j = X_j^E t_j \bmod p, \quad j = 1 ,\ldots, m.$$

5. Compute the second part S of the collective DS:

$$S = S_1S_2\ldots S_m \bmod p.$$

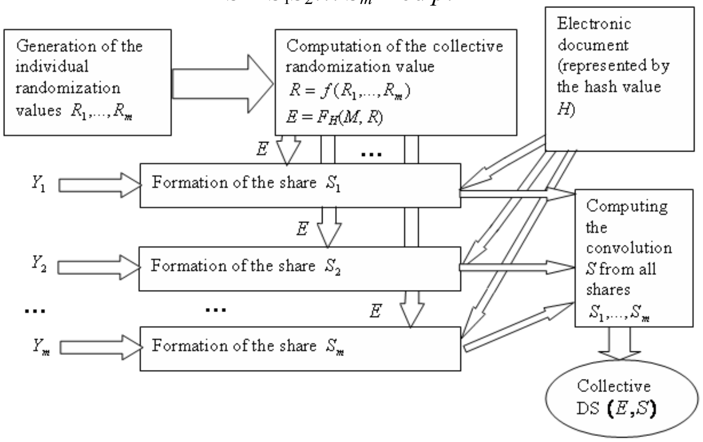
Generation of the individual randomization values $R_1,\ldots,R_m$
Computation of the collective randomization value
$R = f(R_1,\ldots,R_m)$
$E = F_H(M, R)$
Electronic document (represented by the hash value H)
E
...
Y_1
Formation of the share S_1
E
Y_2
Formation of the share S_2
...
...
E
Y_m
Formation of the share S_m
Computing the convolution S from all shares $S_1,\ldots,S_m$
Collective DS (E,S)

Figure 1. Collective DS generation procedure

The CDS verification procedure is performed as follows (see Fig. 2):

1. The collective DS verification procedure uses the collective public key Y that is also computed as a convolution of the set of individual public keys Y_j of all signers:

$$Y = Y_1Y_2\ldots Y_m \bmod p.$$

2. Using the CDS (E, S) compute value R':

$$R^* = S^kY^E \bmod p.$$

3. Compute $E^*= F_H(M, R^*)$.

4. Compare values E' and E. If $E^* = E$, then the signature is valid.

Otherwise the signature is false.

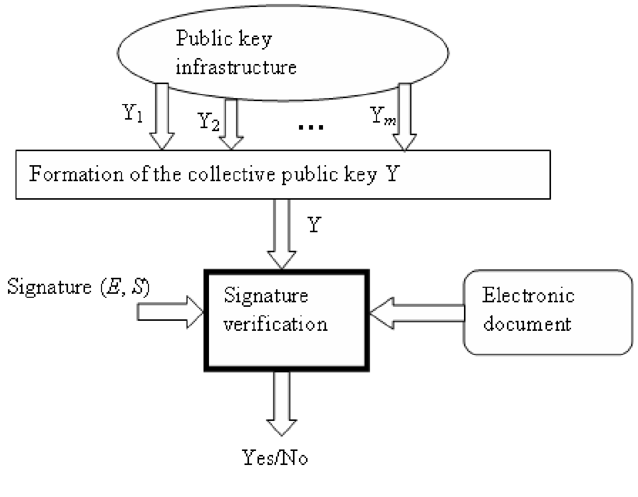

Figure 2. Collective DS verification procedure

In this collective DS the signature length is equal to $|E| + |S| \approx |p|$.

Let us consider attack that is efficient in the case of small size of the compression function $F_H(M, R)$, for example in the case of small value $|\delta|$. It can be performed as the following algorithm:

1. Select at random the values $E < \delta$and $S < p$.

2. Compute value $R^* = S^kY^E \bmod p$.

3. Calculate value $E^*= F_H(M, R')$.

4. Compare values E^* and E. If $E^*\neq E$, then jump to step 1.

On the average the work effort of this algorithm is $\approx 2\delta$ exponentiation operations, since $\Pr(E^*\neq E) = \delta^{-1}$. This attack is infeasible at present, if $F_H(M, R) \geq 160$ bits.

B. Implementation of the Collective DS protocol using standard GOST R 34.10-2001

Using the mechanism of computing convolutions and the general structure of the protocol considered above we have developed the collective DS protocol based on Russian standard GOST R 34.10-2001 [5].

The standard specifies the DS algorithm based on the elliptic curves (ECs) defined over the ground field $GF(p)$ with the following equation:

$$y^2 \equiv x^3 + ax + b \bmod p,$$

where $a,b \in GF(p)$ and y and x are coordinates of the EC points.

For details of the EC cryptography see [6, 7].

Suppose it is given an EC satisfying the requirements by the standard and G is the EC point having large prime order q. Each of m signers generates his private key k_j and his public key $Q_j = k_jG, j = 1 ,\ldots, m$.

The collective DS protocol using GOST R 34.10-2001 is described as follows:

1. Each of signers selects a random value t_j and computes the EC points R_j:

$$R_j = t_jG, \quad j = 1 ,\ldots, m.$$

2. It is computed the following convolution of all points R_j: $R = R_1 + R_2 + \ldots + R_m$ and the value $r = x_R \bmod q$, where x_R is the abscissa of the EC point R. The value r is the first part of the collective DS.

3. Each user computes his share in the collective DS as follows:

$$s_j = (rk_j + t_je) \bmod q,$$

where $e = H \bmod q$ and H is the hash function value computed from the document to be signed.

4. The second part of the signature is:

$$s = s_1 + s_2 +\ldots+ s_m \bmod q.$$

The collective signature is (r, s).

To verify a collective DS one is to perform the following steps:

1. Compute the collective public key as the point

$$Q = Q_1 + Q_2 + \ldots + Q_m.$$

2. Compute the EC point:

$$R^* = (se^{-1} \bmod q)G + ((q - r)e^{-1} \bmod q)Q.$$

3. Compute the value $r^* = x_{R^*} \bmod q$ and compare r^* and r.

If $r^* = r$, then the collective DS is valid.

In this protocol none of the signers generates his individual signature. He generates only its share in the collective DS that corresponds exactly to the given document and to the assigned set of m users. Besides it is computationally difficult to manipulate with shares $s_1, s_2, \ldots, s_m$, and compose another collective DS, relating to some different set of users. This fact imparts on the collective DS the property of the internal integrity, therefore the proposed protocol solves efficiently the problem of signing simultaneously a contract [3].

The proposed protocol possess the following advantages:

i) the digital signature length is sufficiently small and does not depend on number of signers (the collective DS length is equal to the length of individual DS provided by the underlying DS algorithm);

ii) the standard public key infrastructure (PKI) is used;

iii) the protocol is based on the DS algorithm recommended by official state standard;

iv) the protocol can be efficiently used in practice for simultaneous signing a contract;

v) the protocol is as secure as GOST R 34.10-2001 is secure.

The last fact can be proved using the technique applied in [4] to prove security of the collective DS based on standard GOST R 34.10-94 regarding to the following two types of general attacks.

The attack of the first type corresponds to forgery of the collective DS.

The second type attack corresponds to scenario of the calculating the secret key of one of the signers, which shares a collective DS.

In the first attack it is assumed that $m - 1$ legitimate signers attempt to create a collective DS corresponding to m signers.

In the second attack it is assumed that $m - 1$signers that shares some collective DS (r, s) with the mth signer are trying to compute the private key of the mth signer.

It has been proved [4] that any successful method to perform any of the attacks allows breaking the underlying DS algorithm.

III. Composite Signature Protocols

The protocols considered in Section II provide efficient solution of the problem of simultaneous signing a contract. One can generalize the last problem formulating the problem of simultaneous signing a package of contracts, different contracts are being signed by different sets of signers. To solve this extended problem in this section two different variants of the collective digital protocols are proposed. These variants of the collective DS protocols are called composite DS.

A. Variant 1

The first variant of the composite DS scheme uses special verification equation and collective public key computed depending on the hash function values of the documents to be signed. The protocol is based on difficulty of finding discrete logarithm modulo large prime $p = Nq + 1$, where q is a large prime and N is an even number.

Suppose some m users are to sign m different messages $M_1, M_2 \ldots, M_m$ with some single composite DS. Suppose also the jth signer owns the private key $k_j < p$ and the public key $y_j = g^{k_j} \bmod p$, where $j = 1, 2,\ldots, m$.

The composite DS is computed as follows:

1. Each signer participating in the protocol generates the randomization value $R_j = g^{t_j} \bmod p$, where t_j is a random number generated by the jth signer, $j = 1,2, \ldots, m$ and g is a specified number having the order q modulo p.

2. It is computed the convolution of all randomization values R_j, $j = 1,2, \ldots, m$:

$$R = R_1 R_2 \ldots R_m \bmod p.$$

3. It is computed the value e that is the first part of the composite DS:

$$e = R \bmod q.$$

4. Each of signers computes his share in the composite DS using the following formula:

$$S_j = t_j - e h_j k_j \bmod q,$$

where h_j is the hash function value computed from the document M_j to be signed by the jth signer.

5. It is computed the value s that is a convolution of all shares:

$$S = S_1 + S_2 + \ldots + S_m \bmod q.$$

The pair of numbers (e, S) is the composite DS.

The composite DS verification protocol is performed as follows:

1. Compute the collective public key corresponding to the specified set of the signed documents, using the formula:

$$y = y_1^{h_1} y_2^{h_2} y_3^{h_3} \ldots y_m^{h_j} \bmod p,$$

where h_j is the hash function value computed from the document M_j.

2. It is computed the value $R^* = y^e g^S \bmod p$ and $e^* = R^* \bmod q$.

If $e^* = e$, then the composite value is valid.

Peculiarity of this protocol is using special type of the verification equation and computing the collective public key depending on the documents to be signed. There are no DS schemes specified by official standards, which is suitable to implement the composite DS protocol.

B. Variant 2

In order to provide possibility to use the standard GOST R 34.10-2001 as the underlying DS algorithm in the composite DS protocol it is proposed the modification of the second collective DS protocol described in Section II.

There are performed all steps of that protocol, except computing the value e.

In the collective DS protocol this value is computed as $e = H \bmod q$, where H is the hash function value from the given document.

In the composite DS protocol the value e is computed in three steps:

1. Compute the hash function value h_j from each of the documents $M_j, j = 1, 2, \ldots, m$.

2. Compute the has function value from the concatenation of all hash functions h_j and all public keys of the signers

$$H = F_H(h_1||y_1|| \, h_2||y_2||\ldots \, h_m||y_m),$$

where the hash function value h_j is followed by the public key of the jth signer for $j = 1, 2, \ldots, m$.

3. Compute the value $e = H \bmod q$.

This mechanism allows one to transform the given collective DS scheme into the composite DS scheme. In such protocols it is assumed that the public key y_j indicated just after the hash value h_j means that the jth signer agrees to sign exactly the document M_j that corresponds to value h_j.

IV. CONCLUSION

Two new protocols for computing the collective signatures have been proposed.

In the first protocol a set of signers sign the same electronic document. No partial computations can be interpreted as some valid signatures. Therefore the computed collective signature means that all signers have signed the document simultaneously.

In the second protocol the computed composite DS means that some specified subsets of signers have signed different documents simultaneously. Thus the proposed protocols are efficient as solutions of the problems of simultaneous signing a contract and package of contracts, which suites well for practical application.

It is valuable that the protocols can be implemented using the Russian DS standard GOST R 34.10-2001 as the underlying DS scheme.

REFERENCES

[1] Boldyreva A., “Efficient Threshold Signature, Multisignature and Blind Signature Shemes Based on the Gap-Diffi-Hellman-Group Signature Scheme,” LNCS, vol. 2139, pp. 31-46. Springer, Heidelberg 2003.

[2] Min-Shiang Hwang, Cheng-Chi Lee, “Research Issues and Challenges for Multiple Digital Signatures,” International Journal of Network Security, vol. 1, pp. 1-7, 2005.

[3] Schneier B. Applied Cryptography. Second Edition. John Wiley & Sons, Inc. New York, 1996.

[4] Minh N.H., Moldovyan N.A., Minh N.L, “New Multisignature Protocols Based on Randomized Signature Algorithms,” 2008 IEEE International Conference on Research, Innovation and Vision for the Future in computing & Communication Technologies University of Science - Vietnam National University, Ho Chi Minh City, July 13-17, 2008. Proc. PP. 23.pdf, 2008.

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

[6] Koblitz N. Elliptic curve cryptosystems. Mathematics of Computation Advances. 48, 203-209, 1987.

[7] Miller V., “Use of elliptic curves in cryptography,” Advances in cryptology: Proceedings of Crypto'85. LNCS, vol. 218, pp. 417-426. Springer, Heidelberg (1986).