1998 International Computer Symposium
Workshop on Cryptology and Information Security
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

A NOVEL ID-BASED GROUP SIGNATURE

Yuh-Min Tseng, and Jinn-Ke Jan

Institute of Applied Mathematics,
National Chung Hsing University, Taichung, Taiwan 402, ROC
Email: tym@nkjc.edu.tw, jkjan@amath.nchu.edu.tw

ABSTRACT

Group signatures, first introduced by Chaum and Heyst at Eurocrypt'91, allow individual members of a group to make signatures on behalf of the group while providing the signer's anonymity. Most of the previously proposed group signature schemes are based on the discrete logarithm problem, the public keys of users are not identity information, except for the ID-based scheme proposed by Park et al. in 1997. However, Park et al.'s scheme has a serious problem, which is that the all of the previous group signatures signed by other members will be no longer valid if the group is changed. Moreover, the length of the group signature grows linearly with the number of the group members, which make their proposed scheme inefficient. In this paper, the authors propose a novel ID-based group signature scheme which can solve the problem raised by the inclusion of a new group member or the exclusion of an old group member. Meanwhile, compared to Park et al.'s scheme, our scheme requires less computing time for generating the group signature and verifying the group signature. The security of the proposed ID-based group signature scheme is based on the difficulty of computing the discrete logarithm modulo for a composite number.

Keywords: Cryptography, Discrete logarithm,
Group signature, ID-based, Identification

1. INTRODUCTION

In 1991, Chaum and Heyst [1] introduced a new type of signature, called a group signature, that allowed individual members of a group to make signatures on behalf of the group. Furthermore, in the case of a later dispute, the signer can be identified by either the group members together or a group authority. More formally, a group signature scheme has the following three properties:

(i) only members of the group can sign for messages;
(ii) the receiver of the signature can verify that it is a valid signature of that group, but cannot discover which member of the group made it;
(iii) in case of dispute later on, the signature can be "opened" to reveal the identity of the signer.

Chaum and Heyst also presented four schemes that satisfy the above properties. However, every member must choose a new key if the group is changed. In addition, the identification of the signer cannot be identified by the group authority. Their schemes were improved by Chen and Pedersen [2]. However, the group signature schemes that were suggested in [1,2], are interactive and inefficient.

Lee and Chang [6] proposed an efficient noninteractive scheme based on the discrete logarithm problem [3,4]. However, their scheme has the following disadvantage. When the group authority convinces a verifier that the member is indeed the signer, the group authority must announce some extra information to provide the verifier to verify the identity of the signer. But the previous group signatures signed by this signer may be identified by the verifier at the same time. This may be impractical and unreasonable for some applications. In addition, after the announcement, the group authority must renew some keys of the signer. Otherwise, the anonymity of this signer will be exposed in the next signature.

Since the previously proposed schemes [1,2,6] are based on the discrete logarithm problem, the public keys of the signers contain no identity information. That is, every entity needs to select a secret key and compute a corresponding public key. Conventionally, the public keys of users are kept in a public directory. The public keys must be authenticated. A commonly used solution is that every public key is accompanied with a certificate [5] generated by a trusted authority using a digital signature scheme, such as RSA [16] or ElGamal [4]. However, this approach has a potential problem. It wastes both storage in the system and valuable computing time performed by verifiers. In 1984, Shamir [17] first introduced the idea of a cryptographic system based on identification information (e.g., name, address, and physical description). In this system, the public key of each entity is nothing but an identity, which can be defined as part of the identification information.

In 1997, Park et al. [13] presented an ID-based group signature which is based on the Ohta-Okamoto's ID-based signature scheme [12]. The security of the proposed scheme depends on both the discrete logarithm problem and the e th root problem [18]. The group signature is verified from the identities of the group members. However, their scheme has a serious problem in which all of the previous group signatures signed by other members will be invalid if the

group is changed. In addition, the length of a group signature is dependent upon the number of group members.

In this paper, we present a slight improvement on ID-based cryptographic schemes such as Maurer-Yacobi's scheme [8,9] and Lim-Lee's scheme [7]. Based upon the modified ID-based cryptographic scheme, we propose a novel ID-based group signature scheme. The proposed scheme preserves the main merits inherent in the ID-based group signature scheme proposed by Park et al., especially the public key of each entity which is nothing but an identity. Moreover, in the proposed scheme, it solves the problem of the invalidation of previous group signatures caused by the inclusion of a new group member or the exclusion of an old group member. In addition, the group signature is contained to a fixed length.

This paper is organized into six sections. In the next section, the proposed ID-based group signature scheme is presented. In Section 3, we analyze the security of our scheme. Section 4 gives the performance of the proposed scheme. In Section 5, we discuss the properties of our scheme and compare it with the previous work. Finally, a summary is given in Section 6.

2. THE PROPOSED GROUP SIGNATURE SCHEME

The proposed scheme is divided into three stages: the system setup stage, the group signature and verification stage, and the user identification stage.

2.1 System setup stage

The system setup stage consists of two phases: the system initialization phase and the group creation phase. In this subsection, we first present an improvement for the Jacobi symbol method in the Maurer and Yacobi's schemes [8,9].

System Initialization Phase: For the system initialization, the trusted authority chooses two primes p_1 and p_2 of about 120 decimal digits each and $N = p_1 \cdot p_2$, such that the numbers $(p_j - 1)/2$ $(j = 1,2)$ are odd and pairwise relatively prime [10]. Both $(p_1 - 1)/2$ and $(p_2 - 1)/2$ contain several prime factors of 20 decimal digits but no large prime factor. The choice $p_1 \equiv 3 \bmod 8$ and $p_2 \equiv 7 \bmod 8$ is particularly attractive. In this case, the Jacobi symbol $(2/N)$ is -1. With the above limitations for p_1 and p_2, it is easy for the trusted authority to find the discrete logarithms modulo p_1 and p_2 respectively [14,15]. The reader can refer to [7-9] for details.

The trusted authority also selects, once and for all, two integers e and t in $Z^*_{\varphi(N)}$ and computes the corresponding values d and v which are like the parameters in the RSA scheme [16] which satisfies

$$e \cdot d \equiv 1 \bmod \varphi(N),$$

$$v \cdot t \equiv 1 \bmod \varphi(N).$$

, but keeps t, d and v in secret and publishes e. Let g be a primitive element in $GF(p_j)$, for $1 \le j \le 2$. Then the trusted authority computes a public value as

$$F = g^v \bmod N$$

, where $v \equiv t^{-1} \bmod \varphi(N)$. The trusted authority also publishes a one-way function $h(\ )$ which accepts a variable-length input string of bits and produces a fixed-length output string of bits (for example 128 bits) as specified in [11]. The one-way function $h(\ )$ is used to compress the signing message for the ID-based group signature scheme.

When a user U_i (whose identity information is D_i) wants to join the system. The trusted authority computes

$$s_i = e \cdot t \cdot \log_g(ID_i) \bmod \varphi(N)$$

, where

$$ID_i = \begin{cases} D_i \bmod N & if\ (D_i / N) = 1 \\ 2D_i \bmod N & if\ (D_i / N) = -1 \end{cases}$$

In this case, the Jacobi symbol (ID_i / N) will be sure to equal to 1. Therefore, U_i's identity ID_i will be definitely either D_i or $2D_i$ according to the value of the Jacobi symbol (D_i / N). This is guaranteed to have a discrete logarithm. Finally, the trusted authority sends s_i to the user U_i secretly.

Group Creation Phase: Let GA be a group authority with the secret key x and the public key $y \equiv F^x \bmod N$. For each group member U_i with the identity ID_i, the group authority computes

$$x_i \equiv ID_i^{\,x} \bmod N.$$

Then, GA sends x_i to the user U_i secretly.

From the above phases, the system parameters of our scheme are summarized as follows:

(1) the secret values of the trusted authority: p_1, p_2, t, v and d.

(2) the secret values of the trusted authority: N, g, e, F, $h(\ )$.

(3) the secret key of the group authority: x.
(4) the public key of the group authority: y.
(5) the secret key of U_i: s_i and x_i.
(6) the public key of U_i: ID_i.

2.2 Group signature and verification stage

Suppose that a user U_i wants to sign a message M. The group signature generating procedure is as follows. U_i first chooses two random integers r_1 and r_2 in Z_N^*. Then, U_i computes the group signature for message M, i.e., $\{A, B, C, D\}$, where

$$A = y^{r_1} \bmod N$$
$$B = y^{r_2 \cdot e} \bmod N$$
$$C = s_i + r_1 \cdot h(M \| A \| B) + r_2 \cdot e$$
$$D = x_i \cdot y^{r_2 \cdot h(M \| A \| B)} \bmod N$$

where, '||' denotes concatenation. Afterwards, U_i sends $\{M, A, B, C, D\}$ to the verifier.

Upon receiving the messages $\{M, A, B, C, D\}$, any user can verify the group signature for message M as

$$D^e \cdot A^{h(M \| A \| B)} \cdot B \equiv y^C \cdot B^{h(M \| A \| B)} \bmod N$$

If the above equation holds, then the group signature for message M is verified. The correctness of this equation can be easily seen as follows:

$$\begin{aligned}
&D^e \cdot A^{h(M \| A \| B)} \cdot B \bmod N \\
&\equiv (x_i \cdot y^{r_2 \cdot h(M \| A \| B)})^e \cdot (y^{r_1})^{h(M \| A \| B)} \cdot y^{r_2 \cdot e} \bmod N \\
&\equiv x_i^{\,e} \cdot y^{r_2 \cdot h(M \| A \| B) \cdot e} \cdot y^{r_1 \cdot h(M \| A \| B)} \cdot y^{r_2 \cdot e} \bmod N \\
&\equiv (ID_i)^{x \cdot e} \cdot y^{r_2 \cdot h(M \| A \| B) \cdot e} \cdot y^{r_1 \cdot h(M \| A \| B)} \cdot y^{r_2 \cdot e} \bmod N \\
&\equiv (g^{v \cdot d \cdot s_i})^{x \cdot e} \cdot y^{r_2 \cdot h(M \| A \| B) \cdot e} \cdot y^{r_1 \cdot h(M \| A \| B)} \cdot y^{r_2 \cdot e} \bmod N \\
&\equiv (g^{v \cdot x})^{s_i} \cdot y^{r_2 \cdot h(M \| A \| B) \cdot e} \cdot y^{r_1 \cdot h(M \| A \| B)} \cdot y^{r_2 \cdot e} \bmod N \\
&\equiv y^{s_i} \cdot y^{r_2 \cdot h(M \| A \| B) \cdot e} \cdot y^{r_1 \cdot h(M \| A \| B)} \cdot y^{r_2 \cdot e} \bmod N \\
&\equiv y^{s_i + r_1 \cdot h(M \| A \| B) + r_2 \cdot e} \cdot y^{r_2 \cdot e \cdot h(M \| A \| B)} \bmod N \\
&\equiv y^C \cdot B^{h(M \| A \| B)} \bmod N
\end{aligned}$$

2.3 User identification stage

In the case of a dispute, the group signature may be "opened" so that the identity of the signer for the message is revealed. Let us see how a group authority identifies the signer. The group authority with the secret key x can identify the signer without any assistance from group members by finding ID_i satisfying the following equation

$$(ID_i)^{x \cdot e} \equiv D^e \cdot B^{-h(M \| A \| B)} \bmod N$$

for $i = 1 \cdots k$, where k is the number of the group members. The correctness of the above equation can be easily seen as follows:

$$\begin{aligned}
&(ID_i)^{x \cdot e} \equiv (ID_i^{\,x})^e \bmod N \\
&\equiv (x_i)^e \bmod N \\
&\equiv (x_i \cdot y^{r_2 \cdot h(M \| A \| B)} \cdot y^{-r_2 \cdot h(M \| A \| B)})^e \bmod N \\
&\equiv (D \cdot y^{-r_2 \cdot h(M \| A \| B)})^e \bmod N \\
&\equiv D^e \cdot y^{-r_2 \cdot e \cdot h(M \| A \| B)} \bmod N \\
&\equiv D^e \cdot B^{-h(M \| A \| B)} \bmod N
\end{aligned}$$

In order to convince other verifiers that the user U_i with identity ID_i is indeed the signer, the group authority randomly selects an integer r in Z_N^*, and computes

$$R = (ID_i)^{r \cdot e} \bmod N$$
$$S = r + h(M \| A \| B \| R) \cdot x$$

Then the group authority publishes the identification information (R, S) and the user's identity ID_i. Upon receiving the announcement from the authority, the verifier may identify the identity ID_i of the signer for the group signature $\{A, B, C, D\}$ by checking the following equation

$$ID_i^{S \cdot e} \equiv R \cdot (y^C \cdot A^{-h(M \| A \| B)} \cdot B^{-1})^{h(M \| A \| B \| R)} \bmod N$$

If the above equation holds, the user with the identity ID_i is identified. The correctness of the above equation can be easily seen as follows:

$$\begin{aligned}
&ID_i^{S \cdot e} \bmod N \\
&\equiv (ID_i^{\,r} \cdot ID_i^{\,x \cdot h(M \| A \| B \| R)})^e \bmod N \\
&\equiv ID_i^{\,r \cdot e} \cdot ID_i^{\,x \cdot e \cdot h(M \| A \| B \| R)} \bmod N \\
&\equiv R \cdot (ID_i^{\,x \cdot e})^{h(M \| A \| B \| R)} \bmod N \\
&\equiv R \cdot ((g^{v \cdot d \cdot s_i})^{x \cdot e})^{h(M \| A \| B \| R)} \bmod N \\
&\equiv R \cdot (y^{s_i})^{h(M \| A \| B \| R)} \bmod N \\
&\equiv R \cdot (y^C \cdot A^{-h(M \| A \| B)} \cdot B^{-1})^{h(M \| A \| B \| R)} \bmod N
\end{aligned}$$

From the above descriptions, even though the group authority announces (R, S), the group authority need not renew any key for the signer. The reason is that the information (R, S) is only provided for the specific group

signature $\{A,B,C,D\}$ for the message M. The anonymity of any other previous signatures or future signatures for this signer is not damaged.

3. SECURITY ANALYSIS

In Maurer-Yacobi's original scheme [8,9], the prime factors for N were chosen appropriately so that the trusted authority could feasibly compute discrete logarithm modulos for each prime factor. But computing a discrete logarithm modulo N without knowing the prime factors of N is infeasible. It can be shown that computing the discrete logarithm modulo for the composite number N is at least as difficult as factoring the modulus. The system initialization phase of our scheme is like the Jacobi method in the Maurer-Yacobi schemes, expect for an increase of two public values, e and F. Since the modulus N is chosen to be infeasible to factor, specialized attacks applicable to the RSA scheme are ineffective in our scheme, e.g., it is infeasible to find d with the known e. Even though F was published by the system authority, any user can not find prime factors of N.

In the following, some possible attacks against the proposed scheme are presented. Attack 1 concerns that a malicious adversary tries to find the secret key of the group authority. Attacks 2 and 3 are plotted against forgeries, and Attack 4 is plotted against the user anonymity. As we can see, none of these attacks can break our scheme.

Attack 1: An adversary tries to reveal the secret key x of the group authority from either the public key y or the previous announcement (R,S).

The adversary might directly solve x from the equation $y \equiv F^x \bmod N$. In another way, the adversary might solve x from the following equation

$$S = r + h(M||A||B||R) \cdot x$$

This implies that the adversary can obtain x from the above equation if he knows r from $R = (ID_i)^{r \cdot e} \bmod N$. In both these two approaches, however, he will face the difficulty of computing a discrete logarithm modulo for the composite number N, which is at least as difficult as factoring the modulus.

Attack 2: Any legal user U_i of this system and an adversary, who are not group members, may try to forge the group signature.

Since both U_i and the adversary are not group members, they do not have the corresponding secret key x_i. Thus, they cannot generate a valid group signature. As for getting the valid x_i, they might plot the following two approaches: (i) revealing the secret key x of the group authority and then computing $x_i = ID_i^x \bmod N$, (ii) revealing the secret key d of the trusted authority and then computing $x_i = F^{s_i \cdot d} \bmod N$. In the first approach, as analyzed in Attack 1, they will face the computation of the discrete logarithm modulo the composite number N. As for the other approach, since the modulus N is chosen to be infeasible to factor, specialized attacks applicable to the RSA scheme are ineffective, e.g., it is infeasible to find d with the known e.

Attack 3: The group authority or an adversary without the secret key s_i of U_i tries to impersonate U_i and forge the group signature.

Although the group authority knows the secret key x_i of each legal member U_i, he cannot forge the group signature with U_i's identity. The group authority tries to impersonate U_i and forge his signature $\{A,B,C,D\}$ for the message M. The group authority may calculate C satisfying the following verification equation

$$ID_i^{S \cdot e} \equiv R \cdot (y^C \cdot A^{-h(M||A||B)} \cdot B^{-1})^{h(M||A||B||R)} \bmod N.$$

Generating the valid C implies that the group authority must obtain the secret key s_i of U_i. However, the computational complexity to obtain s_i from ID_i or the previous signature of U_i is as difficult as the computation of the discrete logarithm modulo the composite number N. Moreover, since the group authority cannot impersonate U_i and forge the group signature, the adversary's forgery is harder than the group authority's one. Thus, the impersonation attack can not be successful.

Attack 4: A receiver tries to determine the identity of the signer from the group signature.

Each group member can sign a message on behalf of the group without revealing his identity, only the group authority is able to identify the signer. Since the receiver does not know the secret key x of the group authority, he cannot find ID_i satisfying the following equation

$$(ID_i)^{x \cdot e} \equiv D^e \cdot B^{-h(M||A||B)} \bmod N$$

As analyzed in Attack 1, to obtain x implies that he will face the computation of the discrete logarithm modulo the composite number N. That is, the anonymity of the signer in our scheme depends on computing the discrete logarithm modulo the composite number N

4. PERFORMANCES

We first present the computational complexity of the system initialization phase for the proposed scheme. It is like the computational complexity of the Jacobi symbol method in the Muarer-Yacobi's schemes, the computational complexity for the trusted authority only increases the

computing time for the public value F and two multiplication with modulo $\varphi(N)$. However, the trusted authority needs large computing power to compute the secret keys of the users [8,9]. Fortunately, the trusted authority only computes, once and for each user U_i, the discrete logarithm for the corresponding ID_i. Once the system is setup, the trusted authority will not be involved in the process of other group signature stages.

Let us consider the performance of our proposed scheme. The performance evaluation of the proposed scheme concerns the total bit-string length of the group signature and the time complexity. For convenience, the following notations are used to analyze the computational complexity and the transmission cost: $|X|$ is the bit-string length of the message X; T_h is the time for executing the adopted one-way function h; T_{mul} is the time for multiplication without modulo N; T_{mmul} is the time for multiplication with modulo N; $T_{\exp}$ is the time for exponentiation with modulo N. Note that the time for computing addition and subtraction is ignored, because they are much smaller than T_h, T_{mul}, T_{mmul} and $T_{\exp}$.

For practical considerations, we suggested that the group authority may choose his secret key x with 150 bits, while the public parameter e of the trusted authority should be laid between 60-70 bits. The randomly chosen numbers r_1 and r_2 by the signer and r by the group authority are bounded to 160 bits. The fixed-length output of the hash function is 128 bits. Then the size of C in the group signature $\{A,B,C,D\}$ will be bounded to $|\varphi(N)|$. Thus, the size of the group signature is $|A|+|B|+|C|+|D|$, which is bounded to $4|N|$.

For the group signature and verification stage, the time complexity for generating a group signature is the time for computing $\{A,B,C,D\}$ which includes the time to compress the signing message into a digest with shorter length using the one-way hash function $h(\ )$. Therefore, it requires $4T_{\exp}+T_h+2T_{mul}+T_{mmul}$ in total. While upon receiving the group signature, any verifier computes $h(M||A||B)$ and then verifies the group signature. It requires $4T_{\exp}+T_h+3T_{mmul}$.

In the case of a dispute, the group signature must be "opened" to reveal the identity of the signer. As for the group authority, he/she must verify the equation $(ID_i)^{x\cdot e} \equiv D^e \cdot B^{-h(M||A||B)} \bmod N$ for the identity ID_i of each group member. This requires $k\times(4T_{\exp}+T_h+T_{mmul})$, where k is the number of the group members. In order to convince any verifier that the user U_i with the identity ID_i is indeed the signer, the group authority must compute $\{R,S\}$ and publish it. Therefore, it requires $2T_{\exp}+T_h+T_{mul}$ for generating $\{R,S\}$. After receiving the announcement, the verifier may identify the identity of the signer. This requires $6T_{\exp}+2T_h+3T_{mmul}$.

5. DISCUSSIONS AND COMPARISONS

In the previously sections, we have described our ID-based group signature scheme and demonstrated that our scheme has the three properties of the group signatures presented by Chaum and Heyst in [1]. In our scheme, no one can forge the group signature, only legal members of the group can sign messages. Meanwhile, the group authority without the secret key s_i of a legal member U_i is not able to impersonate U_i and forge the group signature. In the case of a later dispute, the identity of the signer for a group signature can be revealed by the group authority. Moreover, in order to convince a verifier that the user with identity ID_i is indeed the signer, the group authority may publish the information (R,S) to enable a verifier to check the identity of the signer, while this does not damage the anonymity of the other previous signatures and any future signatures of this signer. Furthermore, since the secret keys x_i of all legal members U_i computed by the group authority are totally independent, our scheme allows new members to join the group without affecting any other distributed secret keys.

In the ID-based group signature scheme proposed by Park et al., each member computes a group signature in accordance with his secret key and all of the identities of the group members. Therefore, for signing a message on behalf of the group, each signer needs to know all of the identities of the group members while a receiver must also know all of the identities of the group members in order to verify a group signature. That is, the public keys of the group include all of the identities of the group members. For this reason, if the group is changed, group signatures signed using the previous identities of the group members will be invalid. Moreover, the length of the group signature is linear based on the number of group members.

In the following, we make comparisons between the scheme proposed by Park et al.'s in [13] and our proposed scheme. The comparisons are presented in Table 1.

6. CONCLUSIONS

In this paper, we have proposed a new ID-based group signature. We have demonstrated that our scheme has the properties of the group signature, and the group signature is secure against forgeries. Moreover, the proposed scheme allows new group members to join the group dynamically. This solves the problem which occurred in the Park et al.'s scheme. Furthermore, compared to Park et al.'s scheme, the proposed scheme is efficient in terms of computational time and the length of the group signature. We have also demonstrated some possible attacks against the proposed scheme. Under the difficulty of computing discrete logarithms modulo for the composite number N, we have shown that the proposed scheme is secure against these attacks.

REFERENCES

[1] D. Chaum and E. Heyst, Group signatures, in: *Proc. EUROCRYPT'91*, 1992, pp. 257-265.

[2] L. Chen and T. P. Pedersen, New group signature schemes, in: *Proc. EUROCRYPT'94*, 1995, pp. 163-173.

[3] W. Diffie and M. E. Hellman, New directions in cryptography, *IEEE Trans. on Info. Theory*, 22(6):644-654 (1976).

[4] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Trans. Info. Theor.*, 31(4):469-472 (1985).

[5] ISO/IEC 9798-3, Information technology - Security technique - Entity authentication mechanism - Part 3: Entity authentication using a public key algorithm, ISO, 1993.

[6] W. B. Lee and C. C. Chang, Efficient group signature scheme based on the discrete logarithm, *IEE Proc. Comput. Digit. Tech.*, 145(1):15-18 (1998).

[7] C. H. Lim and P. J. Lee, Modified Maurer-Yacobi's scheme and its application, in: *Proc. AUSCRYPT'92*, 1992, pp. 308-323.

[8] U. M. Maurer and Y. Yacobi, Non-interactive public-key Cryptography, in: *Proc. EUROCRYPT'91*, 1992, pp. 498-507.

[9] U. M. Maurer and Y. Yacobi, A non-interactive public-key distribution system, *Designs, Codes and Cryptography*, 9:305-316 (1996).

[10] U. M. Maurer, Fast generation of prime numbers and secure public-key distribution system, *Journal of Cryptology*, 8(3):123-155 (1995).

[11] R. C. Merkle, A fast software one-way hash function, J. Cryptol., 1:43-58 (1990).

[12] K. Ohta and E. Okamoto, Practical extension of Fiat-Shamir scheme, *Electr. Lett.*, 24(15):955-956 (1988).

[13] S. Park, S. Kim, and D. Won, ID-based group signature, *Electr. Lett.*, 33(19):1616-1617 (1997).

[14] S. C. Pohlig and M.E. Hellman, An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance, *IEEE Trans. Info. Theor.*, 24:106-110 (1978).

[15] J. M. Pollard, Theorems on factorization and primality testing, Proc. Cambridge Philos. Society, 76:521-528 (1974).

[16] R. L. Rivest, A. Shamir, and L. Adelman, A method for Obtaining Digital Signature and Public Key Cryptosystem, *Comm. ACM*, 21(2):120-126 (1978).

[17] A. Shamir, Identity-based cryptosystem based on the discrete logarithm problem, in: *Proc. CRYPTO'84*, 1985, pp. 47-53.

[18] M. Stadler, Publicly verifiable secret sharing, in: *Proc. EUROCRYPT'96*, 1997, pp. 190-199.

Table 1. Comparisons of our proposed scheme and the Park et al.'s scheme. "Independent, linear" means that the number is independent respectively linear in the number of group members.

Properties	Proposed scheme	Park et al.'s scheme
Based on assumption	Discrete logarithm modulo the composite number N	Discrete logarithm and e th root problems
Identification of the signer	Group authority	Group authority
Inclusion of new group members	Yes	The previously group signatures signed by other members will be failure
Length of the group's public key	Fixed	Linear
Number of computations during generating a group signature	Independent	Linear
Number of computations during verifying a group signature	Independent	Linear
Length of a group signature	Independent	Linear