A group signature scheme with strong separability

Shundong Xia *, Jinyuan You

Department of Computer Science, Shanghai Jiaotong University, 1954 Huashang Road, Shanghai 200030, People’s Republic of China

Received 28 October 2000; received in revised form 7 March 2001; accepted 4 April 2001

Abstract

Group signatures, introduced by Chaum and van Heijst, allow members of a group to sign messages anonymously on behalf of the group. Only a designated group manager is able to identify the group member who issued a given signature. Many applications of group signatures, for example, electronic market, require that the group manager can be split into a membership manager and a revocation manager. The former is responsible for adding new members to the group. The latter is responsible for opening signatures. Previously proposed group signatures schemes can only achieve a weak form of separability. That is, the revocation manager and the membership manager must work in concert to reveal the identity of the signer. In this paper, we propose a group signature scheme with strong separability in which the revocation manager can work without the involvement of the membership manager. © 2002 Elsevier Science Inc. All rights reserved.

Keywords: Group signature; Strong separability; Identity-based; Discrete logarithm

1. Introduction

Group signatures are a relatively new cryptographic concept introduced by Chaum and Heijst (1992). In contrast to ordinary signatures they provide anonymity to the signer, i.e., a verifier can only tell that a member of some group signed. However, in exceptional cases (such as a legal dispute) any group signature can be “opened” by a group manager to reveal unambiguously the identity of the signature’s originator. At the same time, no one – including the group manager – can misattribute a valid group signature.

Group signatures can be used to conceal organizational structures. For instance, an employee of a large company can use group signatures to sign documents on behalf of the company. In this scenario, it is sufficient for a verifier (who maybe a customer of the company) to know that some representative of the company has signed. Moreover, in contrast to when an ordinary signature scheme would be used, the verifier does not need to check whether a particular employee is allowed to sign contracts on behalf of the company, i.e., he needs only to know a single company’s public key. Other interesting applications of group signatures include e-voting (Nakanishi et al., 1999a), e-bidding (Chen and Pedersen, 1995) and e-cash (Lysyanskaya and Ramzan, 1998; Nakanishi et al., 1999b), etc.

In an ordinary group signature scheme, the group manager has two functions: enabling signers to sign on behalf of the group (managing membership), and revealing the identity of the signature’s originator (managing revocation). However, we argue that it is more flexible to split the group manager into two roles: the membership manager and the revocation manager. The reason for such a splitting is that in many cases, the two roles are better to be played by two different actors. For example, in an electronic market, there are a market administrator, several sellers and several buyers. Group signatures are used to guarantee the anonymity and the safety of transactions. Before providing goods (suppose all goods are information goods, such as digital documents, digital pictures and digital videos, etc.), a seller asks the market administrator to grant him the right to sign on behalf of the market. While sending the goods to a buyer, the seller also attaches the corresponding group signature. The identity of the buyer is only revealed when there is a dispute between the buyer and seller about the goods. The problem is, who has the right to reveal the seller’s identity? If the market administrator has the right, the privacy of sellers will not be protected because that every transaction of sellers can be monitored by the market administrator. A good
solution to this problem is to grant the right to a trusted third part, like a judge. More generally, when the signers want to protect their privacy from the group manager, or when there is a requirement to enhance the security of the group, splitting the group manager is needed.

So group signature schemes with separability are more flexible than ordinary group signature schemes. However, to our best knowledge, previously proposed group signature schemes can only achieve a weak form of separability. That is, if the group manager is split into a membership manager and a revocation manager, the revocation manager and the membership manager must work in concert to reveal the identity of the signer. In other words, if the membership manager refuses to cooperate or provides false information, the revocation manager cannot work correctly. It is unfair to the party who relies on the group signature. For example, in the above electronic market, it can bring damage to the profit of buyers because in case of dispute, the judge maybe having no way to reveal the identity of sellers. So the problem of the separability of group signature scheme is still open.

In this paper, we try to design a scheme with strong separability in which the revocation manager can work without the involvement of the membership manager. Thus the identities of signers are guaranteed to reveal in case of disputes. At the same time, the privacy of signers who behave legally is protected at a high level.

1.1. Preliminaries

Group signature schemes are defined as follows.

**Definition 1.** A group signatures scheme is a digital signature scheme comprised of the following procedures (Ateniese and Tsudik, 1999a):

**SETUP:** An algorithm for generating the initial group public key and a group secret key.

**JOIN:** A protocol between the group manager and a user that results in the user becoming a new group member.

**SIGN:** A protocol between a group member and a user whereby a group signature on a user-supplied message is computed by the group member.

**VERIFY:** An algorithm for establishing the validity of a group signature given a group public key and a signed message.

**OPEN:** An algorithm that, given a signed message and a group secret key, determines the identity of the signer.

A secure group signature scheme must satisfy the following properties (Ateniese et al., 2000):

**Unforgeability:** Only group members are able to sign messages on behalf of the group.

**Anonymity:** Given a valid signature of some message, identifying the actual signer is computationally hard for everyone but the group manager.

**Unlinkability:** Deciding whether two different valid signatures were computed by the same group member is computationally hard.

**Exculpability:** Neither a group member nor the group manager can sign on behalf of other group members.

**Traceability:** The group manager is always able to open a valid signature and identify the actual signer.

1.2. Previous work

Group signatures are first introduced by Chaum and Heijst (1992). They proposed four schemes. Some of schemes do not allow a group manager to add group members after the initial setup. Others require the group manager to contact each member in order to open a signature.

A number of improvements and enhancements followed the initial work. Chen and Pedersen proposed two new schemes in (Chen and Pedersen, 1995; Chen, 1994). These schemes allow the addition of new members after the setup of the system and the distribution of the functionality of the group manager. However, this solution has the drawback that the group manager can falsely accuse a group member of having signed a message. This problem was solved in (Camenisch, 1997; Petersen, 1997). Camenisch and Stadler (1997) proposed the first efficient group signature schemes where the size of the group’s public key and the length of signatures, as well as the computational effort for signing and verifying, are independent of the number of group members. Furthermore, the public key remains unchanged if new members are added to the group. In (Ateniese and Tsudik, 1999b; Ateniese et al., 2000), two more efficient and secure schemes were introduced.

All above schemes can only achieve a weak form of separability. That is, if the group manager is split into a membership manager and a revocation manager, the revocation manager and the membership manager must work in concert to reveal the identity of the signer. Kilian and Petrank (1997, 1998) also indicated this problem. But they did not propose a group signature scheme to solve this problem. In this paper, we try to design a scheme with strong separability in which the revocation manager can work without the involvement of membership manager. Our scheme is based on the idea of identity-based cryptographic system first introduced by Shamir (1985). In such systems, the public key of each entity is nothing but an identity, which can be defined as parts of the identification information (e.g., name, address, and physical description).

In fact, some previous work has already proposed group signatures schemes based on identity-based cryptographic system. However, none of them is prac-
tical. In 1997, Park et al. (1997) presented an ID-based group signature scheme which is based on Ohta-Okamoto’s ID-based signature scheme (Ohta and Okamoto, 1988) and Schoenmaker’s method (Chen and Pedersen, 1995). However, their scheme has a serious problem in which all of the previous group signature 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 1999, Tseng and Jan (1999) proposed a novel ID-based group signature scheme which tried to solve the problem of (Park et al., 1997). Unfortunately, their scheme has been proved (Joye et al., 1999) to be universally forgeable, that is, anyone (not necessarily a group member) is able to produce a valid group signature on an arbitrary message, which cannot be traced by the group manager.

1.3. Our contributions and the organization of the paper

In this paper, we proposed a practical group signature scheme with strong separability such that:

(a) The group manager can be split into a membership manager and a revocation manager. The revocation manager can work without the involvement of the membership manager.

(b) The group public key remains unchanged and previous group signatures remain valid if new members are added to the group.

(c) The lengths of group public key and of the group signature are independent of the number of group members.

(d) The scheme satisfies all the security properties listed in Section 1.1.

The rest of the paper is organized as follows. In Section 2, our basic group signature scheme is presented. In Section 3, we analyze the security of our scheme. In Section 4, we show the strong separability of the scheme. Finally, we discuss and conclude the work of the paper in Section 5.

2. Proposed group signature scheme

The scheme consists of four kinds of participants, a trusted authority for generating secrets keys of all signers, a group manager for managing the memberships and identifying the signers, several signers (group members) for issuing group signatures and several verifiers for checking them. The group manager can also be split into a membership manager (for managing the memberships) and a revocation manager (for identifying the signers).

Generally, the trusted authority can be appointed by the government.

In a company like the example in Section 1, the group manager can be played by the company administrator, the signers can be played by the employees, and the verifiers can be played by the customers.

In an electronic market like the example described in Section 1, it needs to split the group manager. The membership manager can be played by the market administrator, the signers can be played by the sellers, the verifiers can be played by the customers, and the revocation manager can be played by a judge.

This section will introduce our basic scheme. In Section 4, we will show how to split the group manager into a membership manager and a revocation manager.

2.1. Setup of trusted authority

As (Maurer and Yacobi, 1992; Lim and Lee, 1992), the trusted authority generates two prime numbers $p_1$ and $p_2$ of about 100 decimal digits such that both $p_1 - 1$ and $p_2 - 1$ contains several prime factors of 13–15 decimal digits but no larger one and that $(p_1 - 1)/2$ and $(p_2 - 1)/2$ are relatively prime. Let $m = p_1 \cdot p_2$, $p_1$ and $p_2$ can be chosen to satisfy

- $p_1 \equiv \pm 1 \mod 8$ and $p_2 \equiv \pm 3 \mod 8$ so that the Jacobi symbol $(2/m)$ is equal to $-1$.

This case, it is easy for the trusted authority to find the discrete logarithms modulo $p_1$ and $p_2$, respectively (Pohlig and Hellman, 1978; Pollard, 1974). $g$ is chosen such that $g < \min(p_1, p_2)$. Finally, the authority publishes $m$ and $g$ but it keeps secret the prime factors $p_1$ and $p_2$.

2.2. Generating private keys

Because a signer $U_i$’s identity information $D_i$ (which is smaller than $m$) is not guaranteed to have a discrete logarithm modulo a composite number $m$, the trusted authority computes

$$\text{ID}_i = \begin{cases} D_i & \text{if } (D_i/m) = 1, \\ 2D_i \mod m & \text{if } (D_i/m) = -1. \end{cases}$$  \hspace{1cm} (1)$$

In this case, the Jacobi symbol $(\text{ID}_i/m)$ will be sure to equal to 1. Now the trusted authority computes the private key $x_i$ for $U_i$ such that

$$\text{ID}_i = g^{x_i} \mod m.$$  \hspace{1cm} (2)$$

The reader can refer to (Maurer and Yacobi, 1992; Lim and Lee, 1992) for details.

Finally, the trusted authority sends $x_i$ to $U_i$ in a secure way and $U_i$ can check the validity of $x_i$ by verifying Eq. (2).

In order to recover $D_i$ from $\text{ID}_i$, $D_i$ can be composed as follows:

$$D_i = F(\text{identity information}) \| \text{identity information}.$$  

In above equation, ‘\|’ is an operator which concatenates two binary strings. $F$ is a collision-resistant hash function that maps $\{0, 1\}^b$ to $\{0, 1\}^b$, where $b$ is a integer constant. Let $\text{head}(a, c)$ be a function which returns the
first  $c$  bits of binary string  $s$,  \texttt{length}(s)  be a function which returns the length of binary string  $s$, and  \texttt{tail}(a, c)  be a function which returns the last  $c$  bits of binary string  $s$. Then  $D_i$  can be recovered from  $ID_i$  as Eq. (3).

$$D_i = \begin{cases} ID_i, \\ \text{if head}(ID_i, b) \\ \mathcal{H}(\text{tail}(ID_i, \text{length}(ID_i) - b)), \\ \text{if head}(ID_i, b) \\ \neq \mathcal{H}(\text{tail}(ID_i, \text{length}(ID_i) - b)). \\ \end{cases} \quad (3)$$

### 2.3. Setup of group manager

The group manager chooses two large primes  $p_1$  and  $p_4$  such that  $p_3 - 1$  and  $p_4 - 1$  are not smooth,  $n = p_1 \cdot p_4$,  $m < n$, and let  $e$  be an integer satisfying  $\gcd(e, \phi(n)) = 1$.

Next, the group manager chooses two integers  $x \in \mathbb{Z}_n$,  $h \in \mathbb{Z}_n^*$, where  $\mathbb{Z}_n$  denotes the set  $\{0, \ldots, m - 1\}$  and  $\mathbb{Z}_n^*$  denotes the set  $\{a \in \mathbb{Z}_n \mid \text{gcd}(m, a) = 1\}$. Then the group manager computes  $y = h^e \mod m$  satisfying  $y \in \mathbb{Z}_n^*$.

Let  $\mathcal{H}$  be a collision-resistant hash function that maps  $\{0, 1\}^*$  to  $\mathbb{Z}_n$. The public key of the group manager is  $(n, e, h, y, \mathcal{H})$  and his secret key is  $(x, d, p_1, p_4)$, where  $d$  satisfies  $de \equiv 1 \mod \phi(n)$.  $\phi$  is Euler’s totient function and  $\phi(n) = (p_1 - 1) \cdot (p_4 - 1)$.

### 2.4. Generating membership keys

When a signer  $U_i$  wants to join the group, the group manager computes  $ID_i$  according to Eq. (1) and computes

$$z_i = ID_i^d \mod n. \quad \text{(4)}$$

As the membership key of  $U_i$,  $z_i$  is sent to  $U_i$  in a secure way and  $U_i$  checks the validity of  $z_i$  by verifying  $\text{ID}_i = z_i^e \mod n$.

### 2.5. Summary for system parameters

From the above phases, the system parameters of our scheme are summarized as Table 1.

<table>
<thead>
<tr>
<th>Table 1: System parameters of our scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trusted authority</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>Secret values</td>
</tr>
<tr>
<td>Public values</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

### 2.6. Signing phase

To sign a message  $M$,  $U_i$  first chooses five random integers  $x, \beta, \theta, \omega \in \mathbb{Z}_n$, and  $\delta \in \mathbb{Z}_n$. Then  $U_i$  computes

$$A = (y^x \cdot z_i) \mod n,$$

$$B = y^\beta \cdot \text{ID}_i,$$

$$C = h^\omega \mod m,$$

$$D = \mathcal{H}(y^x | g | h | A | B | \tilde{B} | C | \| t_1 \| t_2 \| t_3 | M),$$

where  $\tilde{B} = B \mod m$,  $v = (A^t / B^t) \mod n$,  $t_1 = y^\delta \mod n$,  $t_2 = (y^\delta \cdot g^\delta) \mod m$,  $t_3 = h^\delta \mod m$.

$$E = \delta - D \cdot \varepsilon,$$

where  $\varepsilon = x \cdot e - \omega$.

$$F = \beta - D \cdot \omega,$$

$$G = \theta - D \cdot x_i.$$

Then  $U_i$  sends the $(A, B, C, D, E, F, G)$  to the verifier.

### 2.7. Verification phase

Upon receiving the message  $(A, B, C, D, E, F, G)$, the verifier computes

$$\tilde{B} = B \mod m,$$

$$v' = \left(\frac{A^t}{\tilde{B}}\right) \mod n,$$

$$t_1' = (v'^d \cdot y^\delta) \mod n,$$

$$t_2' = (\tilde{B}^d \cdot y^\delta \cdot g^G) \mod m,$$

$$t_3' = (C^d \cdot h^F) \mod m,$$

$$D' = \mathcal{H}(y^x | g | h | A | B | \tilde{B} | C | \| t_1' \| t_2' \| t_3' | M).$$

The verifier accepts the signature only and only if

$$D' = D. \quad \text{(5)}$$

The correctness of Eq. (5) can be easily seen as follows:

$$t_1' = (v'^d \cdot y^\delta) \mod n$$

$$= (y^x \cdot y^\delta \cdot D^x) \mod n$$

$$= y^\delta \mod n$$

$$= t_1.$$  

$$t_2' = (\tilde{B}^d \cdot y^\delta \cdot g^G) \mod m$$

$$= (y^x \cdot g^y \cdot y^\delta \cdot D^x \cdot g^\delta \cdot D^x) \mod m$$

$$= (y^\delta \cdot g^\delta) \mod m$$

$$= t_2.$$  

$$t_3' = (C^d \cdot h^F) \mod m$$

$$= (h^x \cdot h^\delta \cdot D^x) \mod m$$

$$= h^\delta \mod m$$

$$= t_3.$$  

(6)
2.8. Opening signatures

The group manager recovers $ID_i$ by computing

$$ID_i = \left( \frac{B}{C} \right) \mod m. \tag{8}$$

Then group manager can recover $D_i$ from $ID_i$ according to Eq. (3).

3. Security of the proposed scheme

The security of our scheme is based on the difficulty of the discrete logarithm problem (Camenisch, 1998) and on the security of Schnorr (1991) signature scheme, the RSA (Rivest et al., 1978) signature scheme, ElGamal (1985a,b) encryption scheme. In the following, we show that our scheme satisfies all the security properties listed in Section 1.1.

*Unforgeability and Exculpability*:

If a signer $U_i$ wants to generate a valid group signature, he must have a triple $(ID_i, x_i, z_i)$ which satisfies Eqs. (2) and (4). Since only the trusted authority knows the small prime factors of $m$, it is computationally hard for anybody besides the trusted authority to compute $x_i$ from $ID_i$. Likewise, $p_3$ and $p_4$ are private values of the group manager and only he can easily compute $z_i$ which is the $e$-th root of $ID_i$. The trusted authority does not know $z_i$ and the group manager does not know $x_i$. So only $U_i$ himself has the triple $(ID_i, x_i, z_i)$. It is not feasible for anyone to forge a signature or impersonate $U_i$.

*Anonymity*:

Given a valid signature $(A, B, C, D, E, F, G)$, it is hard for everyone but the group manager to identify the actual signer. All private information are protected by random parameters. ID$_i$ is encrypted in $(B, C)$ and only the group manager knows the decryption key $x$.

*Unlinkability*:

Similarly as *Anonymity*, deciding if two signatures $(A, B, C, D, E, F, G)$ and $(A', B', C', D', E', F', G')$ were generated by the same group member is not feasible.

*Traceability*:

ID$_i$ is encrypted in $(B, C)$ and only the group manager knows the decryption key $x$. By Eqs. (6) and (7), the signer proves that $B \equiv y^{s} \cdot ID_i \mod m$ and $C \equiv h^s \mod m$. So the group manager can get ID$_i$ from Eq. (8).

4. Strong separability of the proposed scheme

The group manager can be split into a membership manager and a revocation manager as follows:

Table 2

<table>
<thead>
<tr>
<th>Secret values</th>
<th>Membership manager</th>
<th>Revocation manager</th>
<th>Signer $U_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1, p_2$</td>
<td>$d, p_3, p_4$</td>
<td>$x$</td>
<td>$x, z_i$</td>
</tr>
<tr>
<td>$m, g$</td>
<td>$n, e, H$</td>
<td>$h, y$</td>
<td>$D, ID_i$</td>
</tr>
</tbody>
</table>

4.1. Membership manager

In the initial phase, the membership manager chooses two large primes $p_3$ and $p_4$ such that $p_3 - 1$ and $p_4 - 1$ are not smooth, $n = p_3 \cdot p_4$, $m < n$, and let $e$ be an integer satisfying $gcd(e, \phi(n)) = 1$. Let $H$ be a collision-resistant hash function that maps $\{0, 1\}^*$ to $\mathbb{Z}_m$. The public key of the membership manager is $(n, e, H)$ and his secret key is $(d, p_3, p_4)$, where $d$ satisfies $de \equiv 1 \mod \phi(n)$.

Then the membership manager can generate the membership keys for group members as the method described in Section 2.4.

4.2. Revocation manager

In the initial phase, the revocation manager chooses two integers $x \in \mathbb{Z}_m$, $h \in \mathbb{Z}_m^*$ and computes $y = h^x \mod m$ satisfying $y \in \mathbb{Z}_m^*$. Then the revocation manager publishes $h$ and $y$ but it keeps secret $x$.

In the working phase, the revocation manager identifies the signer as the method described in Section 2.8. Notice that the revocation manager does not need any help from the membership manager in this phase.

4.3. Summary of system parameters

The system parameters of the extended scheme can be summarized as Table 2.

5. Discussion and summary

Previously proposed practical group signature schemes are based on the discrete logarithm problem, the public keys of the signers contain no identity information. That is, every signer needs to select a secret key and compute a corresponding public key. Thus the membership manager must maintain a directory which maps the signers’ public keys to their identity information. While in the phase of opening signatures, the revocation manager can only reveal the public keys of the signers and has to ask the membership manager to provide the corresponding identity information. If the membership manager refuses to cooperate or provide false information, the revocation manager cannot get the true identity information of the signers. So only a weak form of separability can be achieved in these schemes.
Our scheme is based on the idea of identity-based cryptographic system first introduced by Shamir (1985). In our scheme, the public key of each signer is nothing but identity information. And in the phase of verification, the verifiers can make sure that the group signatures really contains the identity information of signers though the verifiers have no way to reveal the identity information themselves. Then in the phase of opening signatures, the revocation manager can successfully get the identity information of signers without the involvement of the membership manager. So our scheme has the feature of strong separability.

Besides, in our scheme, the group public key and previous signature remains valid when a new member is added to the group. Furthermore, the lengths of group public key and of signature are independent of the number of group members. We have also demonstrated that our scheme satisfies the necessary security properties.

References


Shandong Xia received the B.S. and M.S. degrees in Computer Science from Nanjing University of Aeronautics & Astronautics in 1996 and 1999, respectively. He is presently a Ph.D. candidate in Department of Computer Science, Shanghai Jiao Tong University. His research interests include cryptography, network computing, etc.

Jinyuan You is the chairman of the Department of Computer Science, Shanghai Jiao Tong University. His research interests include operating system, information security, distributed systems, etc.