DE GRUYTER DOI 10.1515/jisys-2012-0024 — Journal of Intelligent Systems 2013; 22(2): 95–110

Haiyong Bao* and Zhenfu Cao

Group–Proxy Signature Scheme: A Novel Solution to Electronic Cash

Abstract: Proxy signature and group signature are two basic cryptographic primitives. Due to their valuable characteristics, many schemes have been put forward independently and they have been applied in many practical scenarios up to the present. However, with the development of electronic commerce, many special requirements come into being. In this article, we put forward the concept of group–proxy signature, which integrates the merits of proxy signature and group signature for the first time. We also demonstrate how to apply our scheme to construct an electronic cash system. The space, time, and communication complexities of the relevant parameters and processing procedures are independent of group size. Our demonstration of the concrete group–proxy signature scheme shows that the concepts brought forward by us are sure to elicit much consideration in the future.

Keywords: Digital signature, group signature, proxy signature, group–proxy signature.

***Corresponding author: Haiyong Bao,** College of Computer and Information Engineering, Zhejiang Gongshang University, No. 18, Xuezheng Street, Hangzhou 310018, China, e-mail: baohaiyong1@163.com
Zhenfu Cao: Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China

1 Introduction

In a proxy signature scheme, an original signer can delegate his sign ability to an entity called proxy signer. Then, the proxy signer can sign messages on behalf of the original signer. Since Mambo et al. [18, 19] put forward the concept of proxy signature [18, 19], many proxy signature schemes have been proposed [8, 11, 13, 15, 16, 22, 24–26].

The group signature scheme, introduced by Chaum and van Heyst [10], allows one user from a group to sign messages on behalf of the group without revealing his identity and without allowing the messages to be linkable to other signed messages. Thus, a group signature can be verified with the same and unique public key of the group. There exists an identity called *group manager* who can “open” particular signatures to revoke the anonymity of the signatures in such schemes. Sometimes a *group manager* can be divided into an issue manager called *issuer* and an open manager called *opener*. This is a desirable property that allows the distribution of trust. Group signatures are among the most important cryptographic primitives for providing privacy and have been used for applications such as anonymous credentials, identity escrow, voting and bidding, electronic cash (e-cash), etc. Many group signature schemes [1–7, 12, 14] have also been put forward as yet.

Let us consider the following scenario. Let O be the root entity of a company or an organization. For some reasons, for instance, it is absent or to improve processing efficiency, the root entity O will delegate his sign ability to its sub-entities similar to the proxy signature scheme. Let P be the set of the sub-entities and P_i be the entity with identity $i \in P$. Then, each sub-entity P_i can sign messages on behalf of the original signer O. Under some circumstances, for example, the signer’s privacy needs to be protected, then any verifier can check the validity of the signature similar to the ordinary signature schemes but cannot compromise the real identity corresponding to the signature. Many branch banks who can dispense e-cash on behalf of the country’s central bank can be regarded as the typical application of the above scenario. Such schemes should have the following properties [9, 17]:

1. No bank should be able to trace any e-cash it issued. Therefore, the consumers can spend their e-cash anonymously.
2. All the branch banks form a group. Any vendor only needs to invoke a single universal verification procedure, based on the group public key, to ensure the validity of any e-cash he receives.
3. There is a single public key for the entire group of the branch banks, and the size of this public key should be independent of the number of the branch banks. Therefore, when more branch banks join the group, the public key does not need to be modified.
4. In case of dispute, there exists some entity in the group formed by the branch banks who can “open” the particular signature and trace the real branch bank who issued the particular e-cash. No one else can determine the identity. No vendor can even determine the bank from which the consumer obtained his e-cash even though the vendor can easily check that the e-cash is valid. This restriction gives an extra layer of anonymity because it conceals both the spender’s identity and the bank who issued the e-cash.

One key point of realizing multibank e-cash system is *secure delegation*. Concretely, the central bank securely delegates his sign ability to the branch banks. It is similar to the concept of proxy signature. Thus, the concept of proxy signature can be applied to the e-cash system.

Another key point of realizing multibank e-cash system is *anonymity*. Specifically, after being delegated by the central bank, each branch bank should issue the e-cash anonymously. However, to resist malicious branch banks, there also exists a group manager in the group formed by all the branch banks who can "open" the particular signature and trace the actual issuer. Thus, from this angle, the concept of anonymous e-cash system is similar to the concept of group signature, and the ideas of group signatures can be adopted in e-cash system.

Some scheme [27] just adopts the proxy signature to realize the e-cash system. To prevent the malicious original signer, a third trust party is introduced into the system. Thus, the efficiency is compromised. Furthermore, to satisfy the anonymity requirements, the identity of the branch bank cannot be used to verify the validity of the ultimate signatures. Thus, to satisfy the above requirements, only proxy-unprotected signature can be adopted. That is, all the branch banks use the same secret key delegated jointly by the central bank and the trust party to generate the signatures. Thus, on the one hand, schemes constructed this way cannot resist a frame attack coming from the coalition of the central bank and the third trust party. On the other hand, to resist the coalition attack, the branch bank's secret key could be additionally required and integrated into the delegation key generated by the central bank and the third party. However, when verifying the ultimate signature, the branch bank's public key is needed. Thus, a large number of the branch bank's public key certificates need to be stored, and efficiency declines quickly. Furthermore, the anonymity is difficult to satisfy when the branch bank's private key and public key are used to generate and verify the signature, respectively.

Lysyanskaya and Ramzan [17] combined the notions of blind signatures and group signatures and presented a group–blind signature and applied it to e-cash system. However, first, their scheme is of low efficiency because the vendor must engage in a protocol with the bank to resist the double spending each time he receives an e-coin from the consumer. Second, the consumer must engage in an expensive several round interactive protocol with the banks each time he wants a new e-coin.

To achieve the requirements of a secure e-cash system listed above, we put forward a novel practical group–proxy signature scheme. Our scheme combines the already existing notions of proxy signatures and group signatures. It is an extension of Miyaji and Umeda's signature scheme [20] that adds the *proxy* property. We will show how our scheme can be applied to construct a practical e-cash system where multiple branch banks pertained to the central bank and the branch banks can substitute the central bank to distribute anonymous and untraceable e-cash. Moreover, the identity of the e-cash-issuing bank, i.e., the branch bank, is concealed. All the branch banks form a group, and in case of dispute, the anonymity of the signature can be compromised by the group manager entity in the group.

The rest of the article is organized as follows. In the next section, we describe some preliminaries and the concept and model of the group–proxy signature. Then, our concrete group–proxy signature scheme is presented in Section 3. Some security analysis and performance evaluations are discussed in Sections 4 and 5, respectively. Finally, the conclusions and remarks are made in Section 6.

2 Preliminaries and the Group–Proxy Signature Model

2.1 Notations

For a set A, $a \in_R A$ means that a is chosen randomly and uniformly from A, and $A \setminus \{a\}$ means that $A - \{a\} = \{x \in A | x \neq a\}$. For a group $G = \langle g \rangle$, $ord(g)$ means order of g in G. The bit length of a is denoted by $|a|$. Let $c[j]$ be the j^{th} bit of a string c. We use $\|$ to denote the concatenation of two (binary) strings or of binary representations of integers and group elements.

2.2 Signature Based on Proof of Knowledge

In a signature based on proof of knowledge (SPK), the signer ties his knowledge of a secret to the message to be signed. Thus, a signature of knowledge is used to sign a message and to prove the knowledge of the secret. Without the secret, no one can generate the valid signature. Signatures of knowledge were used by Camenisch and Stadler [6]. Their construction is based on the Schnorr signature scheme [23] to prove knowledge. All the signature of knowledge proposed by Camenisch and Stadler [6] can be proved secure in the random oracle model and their interactive versions are zero-knowledge.

We use several existing protocols of signature of knowledge [6] such as proving the knowledge of double discrete logarithm on a message and proving the knowledge of representations of some elements to the other bases on a message as building blocks in our proposed scheme. Due to space limitation, we omit the details of these protocols here. For more discussions, please refer to Ref. [6].

2.3 The Group–Proxy Signature Model

Our group–proxy signature model extends the group signature over only known-order group [20]. Generally speaking, our proposed group–proxy signature model comes from the basic group signatures [20] and integrates the properties of proxy signatures. Our model allows the member of a group to sign messages anonymously on behalf of some original entity such that the following properties hold for the resulting signature.

– Strong unforgeability: Only delegated group members can issue valid signatures on behalf of the original signer.
– Signer anonymity and signatures unlinkability: Given a signature, no one except the group manager can determine which group member issued the signature. It is infeasible to determine whether two message–signature pairs were issued by the same group member.
– Undeniable signer identity: The identity of the group member who issued the valid signature can always be traced by the entity called group manager.
– Secure against framing attacks: Neither the original signer nor the coalition of the original signer and some subset of the delegated group members can generate valid signatures on behalf of other group members.

2.4 Participants and Procedures

Our group–proxy signature scheme consists of a trusted party for the initial setup, the original signer O, two group managers (the *issuer* and the *opener*), and group users P_i with unique identities $i \in N$ (the set of positive integers). All P_i's form a group G. Each P_i can join G by its interaction with the *issuer*. Then, P_i can sign messages anonymously on behalf of O. In case of dispute, the identity of each group–proxy signature can be traced by the *opener*. Our scheme is specified as a tuple $GPS = (Setup, Delegate, Join, Sign, Verify, Open)$ of polynomial–time algorithms described as follows.

– Setup: A probabilistic algorithm that generates the public keys Y_o, Y_{ik}, Y_{ok} and the corresponding private keys x_o, x_{ik}, x_{ok} for the original signer O, the *issuer*, and the *opener*.
– Delegate: An algorithm that inputs the original signer's secret key and outputs the proxy signer P_i's delegation key s_o.
– Join: An interactive protocol between the *issuer* and the new group member Alice that produces Alice's secret key x, her membership certificate v, and her public key z.

- Sign: An algorithm on input message m, some user's delegation key s_o, secret key x, and his membership certificate v produces a signature s on m that satisfies the properties above.
- Verify: An algorithm on input $(m, s, Y_o, Y_{ik}, Y_{ok})$, determines if s is a valid signature for the message m with respect to the corresponding public information.
- Open: An algorithm on input (s, x_{ok}) returns the identity of the real identity who issued the signature s.

3 Our Proposed Scheme

As shown in the above sections, our proposed group–proxy signature scheme extends the group signature scheme [20] that adds the *proxy* property. Our scheme allows P_i to prove the knowledge of his delegation key s_o corresponding to the valid delegation procedures and the original signer's private key and his membership certificate (A_i, b_i) corresponding to his secret key x_i. Let $\tilde{P}$ be a prime with $P \mid \tilde{P} - 1$, $P = pq$ and $q \mid p - 1$. Let g, g_1, g_2 be the q-order elements in Z_P^*. Thus, after obtaining a delegation key s_o from the original signer O, P_i selects a secret key x_i and obtains his membership certificate (A_i, b_i), satisfying $A_i = g_2^{x_i} g^{s_o} g_1^{-b_i} y_{ik}^{-A_i} \bmod P$. Precisely, our proposed scheme consists of six algorithms as follows.

3.1 System Setup

Suppose k is a security parameter. To set up the group–proxy signature scheme, similar to Miyaji and Umeda [20], the following procedures are performed:

1. Suppose $H : \{0, 1\}^\star \to \{0, 1\}^k$ be the secure, collision resistant, one-way hash function.
2. Choose a random k-bit prime q and a random prime p, such that $q \mid (p - 1)$ and set $P = pq$.
3. Choose a random prime $\tilde{P}$ such that $P \mid (\tilde{P} - 1)$.
4. Denote cyclic subgroup $G_P \subset Z_P^\star$ with order q and $G_{\tilde{P}} \subset Z_{\tilde{P}}^\star$ with order P.
5. Choose random elements g, h, g_1, g_2, g_3 and $g_4 \in_R G_P \setminus \{1\}$ such that the discrete logarithms based on each other elements are unknown.
6. Choose a random element $\tilde{g} \in_R G_{\tilde{P}} \setminus \{1\}$.
7. Compute $y_o = g^{x_o} \bmod P$, $y_{ik} = g_1^{x_{ik}} \bmod P$, and $y_{ok} = g_3^{x_{ok}} \bmod P$, where x_o, x_{ik}, $x_{ok} \in_R Z_q$ are kept secret by the original signer O, the *issuer* and the *opener*, respectively.

8. Output the group public key $PK = (q, P, \tilde{P}, g, h, g_1, g_2, g_3, g_4, \tilde{g}, y_o, y_{ik}, y_{ok})$ and the secret key $SK = (x_o, x_{ik}, x_{ok} \in_R Z_q)$.

3.2 Generation of the Delegation Key

The original signer O performs the following steps to generate the proxy signing key and delegate his sign ability to the proxy signers:

1. Generate the warrant m_ω, which consists of the type of the delegated information, the original and the proxy signers' identities, the delegation period, etc.
2. Select $k_o \in_R Z_q$ and compute

$$r_o = g^{k_o} \bmod P,$$

$$s_o = x_o H(m_\omega \,||\, r_o) + k_o \bmod q,$$

$$S_o = g^{s_o} \bmod P,$$

$$T_o = H\left(m_\omega \,||\, r_o\right)^{s_o} \bmod P.$$

   r_o, S_o, T_o are published by the original signer.
3. Send (r_o, s_o, m_ω) to the proxy signer $P_i \in G$ via secure channel.

After receiving (r_o, s_o, m_ω), the proxy signer P_i verifies the validity of the original signer O's signature by checking

$$S_o = g^{s_o} = y_o^{H(m_\omega \,||\, r_o)} r_o \bmod P.$$

Finally, s_o is kept secret by the proxy signer P_i, and it will be used to generate the proxy signer's *private signing key* in the later sections.

3.3 Registration

In this protocol, after receiving the delegation secret s_o, user P_i and the *issuer* jointly perform the following procedures. In the end, P_i obtains a random value $x_i \in Z_q$ satisfying $A_i = g_2^{x_i} g^{s_o} g_1^{-b_i} y_{ik}^{-A_i} \bmod P$, where x_i is only known by P_i, and A_i, b_i are generated by the issuer manager. The concrete procedures are as follows.

1. $P_i \rightarrow issuer : I = g_2^t h^r \bmod P$, where $t, r \in_R Z_q$.
2. $P_i \leftarrow issuer : u, v \in_R Z_q$.
3. P_i computes $x_i = ut + v,\ y_i = g_2^{x_i} g^{s_o} \bmod P$.

4. $P_i \rightarrow issuer : y_i$ and a signature on the warrant m_ω based on the proof of knowledge of (α, β, τ) such that $T_o = H(m_\omega \parallel r_o)^\beta \bmod P$, $y_i = g_2^\alpha g^\beta \bmod P$, and $S_o g_2^v I^u = h^\tau y_i \bmod P$. Precisely, P_i executes as follows.
   (a) Choose random integers $r_1, r_2, r_3 \in_R Z_q$.
   (b) Compute $T = S_o g_2^v I^u y_i^{-1} \bmod P$ and $t_o = H(m_\omega \parallel r_o) \bmod P$ and $T_1 = g_2^{r_1} g^{r_3} \bmod P$, $T_2 = h^{r_2} \bmod P$, $T_3 = t_o^{r_3} \bmod P$.
   (c) Compute $c = H(g_2 \parallel g \parallel h \parallel t_o \parallel y_i \parallel T \parallel T_o \parallel T_1 \parallel T_2 \parallel T_3 \parallel r_o \parallel m_\omega)$.
   (d) Compute in Z_p: $s_1 = r_1 - cx_i$, $s_2 = r_2 - cur$, $s_3 = r_3 - cs_o$.
   (e) Send (c, s_1, s_2, s_3) to the *issuer*.

   After receiving (c, s_1, s_2, s_3), the *issuer* computes

$$t_o = H\left(m_\omega \| r_o\right) \bmod P,\ T = S_o g_2^v I^u y_i^{-1} \bmod P$$

$$c' = H\left(g_2 \| g \| h \| t_o \| y_i \| T \| T_o \| g_2^{s_1} g^{s_3} y_i^c \| h^{s_2} T^c \| t_o^{s_3} T_o^c \| r_o \| m_\omega\right)$$

   and checks whether $c' = c$ holds. If not, the protocol aborts.
5. $U_i \leftarrow issuer : A_i = y_i g_1^{-w_i} \bmod P$ and $b_i = w_i - A_i x_{ik} \bmod q$, where $w_i \in_R Z_q$.
6. U_i verifies that $g_2^{x_i} g^{s_o} = A_i g_1^{b_i} y_{ik}^{A_i} \bmod P$.

Finally, P_i obtains a membership key (x_i, s_o) and a membership certificate $(A_i, b_i) \in Z_P \times Z_q$, and the *issuer* inserts (ID_i, A_i, b_i) into the member list *ML*.

3.4 Signature Generation

To sign a message m, any legitimate P_i executes the following procedures.
1. Choose a random integer $w \in_R Z_q$.
2. Compute $T_1 = \tilde{g}^{g_3^w} \bmod \tilde{P}$, $T_2 = T_1^{g_4^{b_i}} \bmod \tilde{P}$, $T_3 = g_3^{b_i} g_4^w g_2^{x_i} \bmod P$ and $T_4 = A_i g_3^w \bmod P$, $T_5 = y_{ok}^w \bmod P$ and $T_6 = T_2^{g_3^{x_i}} \bmod \tilde{P}$.
3. Choose random integers $w_{1j}, w_{2j}, w_{3j} \in_R Z_q$, for $1 \le j \le k$.
4. Compute $t_{1j} = \tilde{g}^{g_3^{w_{2j}}} \bmod \tilde{P}$, $t_{2j} = T_1^{g_4^{w_{1j}}} \bmod \tilde{P}$, $t_{3j} = T_2^{g_3^{w_{3j}}} \bmod \tilde{P}$, and $t_{4j} = g_3^{w_{1j}} g_4^{w_{2j}} g_2^{w_{3j}} \bmod P$, for $1 \le j \le k$.
5. Compute
   $c_1 = H(g_2 \parallel g_3 \parallel g_4 \parallel \tilde{g} \parallel T_1 \parallel T_2 \parallel T_3 \parallel T_6 \parallel t_{11} \parallel \ldots \parallel t_{1k} \parallel t_{21} \parallel \ldots \parallel t_{2k} \parallel t_{31} \parallel \ldots \parallel t_{3k} \parallel t_{41} \parallel \ldots \parallel t_{4k} \parallel m_\omega \parallel m)$,
   $s_{1j} = w_{1j} - c_1[j] b_i \bmod q$, $s_{2j} = w_{2j} - c_1[j] w \bmod q$, and $s_{3j} = w_{3j} - c_1[j] x_i \bmod q$, for $1 \le j \le k$.

Remark 1. The above protocol is actually the signature of knowledge of

$$SPK\Big\{(\alpha_1, \alpha_2, \alpha_3) : T_1 = \tilde{g}^{g_3^{\alpha_2}} \bmod \tilde{P} \wedge T_2 = T_1^{g_4^{\alpha_1}} \bmod \tilde{P}$$
$$\wedge T_3 = g_3^{\alpha_1} g_4^{\alpha_2} g_2^{\alpha_3} \bmod P \text{ and } T_6 = T_2^{g_3^{\alpha_3}} \bmod \tilde{P}\Big\}(\ ),$$

and the set of $(c_1, s_{11}, \ldots, s_{1k}, s_{21}, \ldots, s_{2k}, s_{31}, \ldots, s_{3k}) \in \{0, 1\}^k \times Z_q^{3k}$ is the corresponding signature.

Subsequently, P_i executes the following procedures.

1. Choose random integers $w_4 \in_R Z_P$, w_5, w_6, w_7, $w_8 \in_R Z_P$.
2. Compute $t_o = H(m_\omega \parallel r_o) \bmod P$.
3. Compute $t_5 = g_3^{w_5} g_4^{w_7} g_2^{w_6} \bmod P$, $t_6 = y_{ik}^{-w_4} g_1^{-w_5} g_2^{w_6} g_3^{w_7} g^{w_8} \bmod P$, $t_7 = y_{ok}^{w_7} \bmod P$, $t_8 = T_1^{w_4} \bmod \widetilde{P}$, and $t_9 = t_o^{w_8} \bmod P$.
4. Compute
   $c_2 = H(g_1 \parallel g_2 \parallel g_3 \parallel g_4 \parallel g \parallel \tilde{g} \parallel y_{ik} \parallel y_{ok} \parallel T_1 \parallel T_3 \parallel T_4 \parallel T_5 \parallel T_o \parallel t_5 \parallel t_6 \parallel t_7 \parallel t_8 \parallel t_9 \parallel m_\omega \parallel m)$,
   $s_4 = w_4 - c_2 A_i \bmod P$, $s_5 = w_5 - c_2 b_i \bmod q$, $s_6 = w_6 - c_2 x_i \bmod q$, $s_7 = w_7 - c_2 w \bmod q$, and $s_8 = w_8 - c_2 s_o \bmod q$.

Remark 2. The above protocol is actually the signature of knowledge of

$$\begin{aligned} SPK\Big\{(\alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8) : \alpha_4 \in Z_P \wedge T_3 = g_3^{\alpha_5} g_4^{\alpha_7} g_2^{\alpha_6} \bmod P \\ \wedge T_4 = y_{ik}^{-\alpha_4} g_1^{-\alpha_5} g_2^{\alpha_6} g_3^{\alpha_7} g^{\alpha_8} \bmod P \\ \wedge T_5 = y_{ok}^{\alpha_7} \bmod P \wedge \tilde{g}^{T_4} = T_1^{\alpha_4} \bmod \tilde{P} \\ \text{and } T_o = t_o^{\alpha_8} \bmod P\Big\}(\ ), \end{aligned}$$

and the set of $(c_2, s_4, s_5, s_6, s_7, s_8) \in \{0, 1\}^k \times Z_p \times Z_q^4$ is the corresponding signature.

3.5 Signature Verification

Any verifier can perform the following procedures to check the validity of the signature.

1. Compute

$$t'_{1j} = \begin{cases} \tilde{g}^{g_3^{w_{2j}}} \bmod \tilde{P} \text{ if } c_1[j] = 0 \\ T_1^{g_3^{s_{2j}}} \bmod \tilde{P} \text{ otherwise} \end{cases},$$

$$t'_{2j} = \begin{cases} T_1^{g_4^{w_{1j}}} \bmod \widetilde{P} \text{ if } c_1[j] = 0 \\ T_2^{g_4^{s_{1j}}} \bmod \widetilde{P} \text{ otherwise} \end{cases},$$

$$t'_{3j} = \begin{cases} T_2^{g_3^{w_{3j}}} \bmod \widetilde{P} \text{ if } c_1[j] = 0 \\ T_6^{g_3^{s_{3j}}} \bmod \widetilde{P} \text{ otherwise} \end{cases},$$

$$t'_{4j} = \begin{cases} g_3^{w_{1j}} g_4^{w_{2j}} g_2^{w_{3i}} \bmod P \text{ if } c_1[j]=0 \\ T_3 g_3^{s_{1j}} g_4^{s_{2j}} g_2^{s_{3i}} \bmod P \text{ otherwise} \end{cases},$$

and

$$c'_1 = H\big(g_2 \| g_3 \| g_4 \| \tilde{g} \| T_1 \| T_2 \| T_3 \| T_6 \| t'_{11} \| \ldots \| t'_{1k} \| t'_{21} \| \ldots \| t'_{2k} \| t'_{31} \| \ldots \\ \| t'_{3k} \| t'_{41} \| \ldots \| t'_{4k} \| m_\omega \| m\big).$$

2. Compute $t'_5 = g_3^{s_5} g_4^{s_7} g_2^{s_6} T_3^{c_2} \bmod P$, $t'_6 = y_{ik}^{-s_4} g_1^{-s_5} g_2^{s_6} g_3^{s_7} g^{s_8} T_4^{c_2} \bmod P$, $t'_7 = y_{ok}^{s_7} T_5^{c_2} \bmod P$, $t'_8 = T_1^{s_4} (\tilde{g}^{T_4})^{c_2} \bmod \tilde{P}$, $t'_9 = t_o^{s_8} T_o^{c_2} \bmod P$, and $c'_2 = H\big(g_1 \| g_2 \| g_3 \| g_4 \| g \| \tilde{g} \| y_{ik} \| y_{ok} \| T_1 \| T_3 \| T_4 \| T_5 \| T_o \| t'_5 \| t'_6 \| t'_7 \| t'_8 \| t'_9 \| m_\omega \| m\big)$.
3. Check whether $c'_1 = c_1$ and $c_2 = c'_2$ hold. If it does, the signature is valid. Otherwise, it will be rejected.

3.6 Open

To trace the real identity of a signature on message m, the *opener* executes as follows.
1. Check the validity of the signature on message m.
2. Compute $A_i = T_4 / T_5^{1/x_{ok}} \bmod P$.
3. Identify a signer P_i from A_i using the member list ML.
4. Output the signer's identity ID_i.

4 Security of the Proposed Scheme

Our proposed group–proxy signature scheme extends Miyaji and Umeda's [20] basic group signature scheme and adds the *proxy* property. In this section, we will show that our revised *group signature* part is secure as of old. Furthermore, our security analysis indicates that the added *proxy signature* part is secure as well.

4.1 Security Proof of the Delegation Phase

As presented in Section 3.2, the proxy signer $P_i \in G$ obtains (r_o, s_o, m_ω) from the original signer O. Actually, (r_o, s_o) is the original signer O's signature on the warrant m_ω. The correctness of the signature is obvious. In fact, we use Schnorr-type signature [21], which has been proved secure in the random oracle model, to generate such signature. In general, the warrant m_ω contains the type of the delegated information, the original and the proxy signers' identities, the delegation period, etc. Thus, from the angle of the *proxy signature*, neither can the delegated proxy signer $P_i \in G$ abuse the rights delegated by the original signer nor can anyone forge a valid delegation tuple. In the later sections, we will show that any verifier can be convinced that the valid group–proxy signature implies that the proxy signer P_i owns the original signer O's authorization.

4.2 Security Proof of the Membership Certificate

Similar to Miyaji and Umeda [20], the security of our revised membership certification-generation algorithm is based on the difficulty of the strong multiple discrete logarithm problem (MDLP).

For the security parameter k, let p and q be primes with $|q| = k$ and $q | p - 1$. Set $P = pq$, and let $g_1, g_2, g_3, g_4 \in Z_P^*$ be elements with order q. Define a set $\chi(Z_P, g_1, g_2, g_3, g_4) = \{(x_1, x_2, x_3, x_4) \in Z_P \times Z_q^3 | x_1 g_1^{x_1} g_2^{x_2} = g_3^{x_3} g_4^{x_4} \bmod P\}$, where the discrete logarithms of g_1, g_2, g_3, and g_4 based on each other element are not known.

Problem (strong MDLP). Given Z_P, g_1, g_2, g_3 and $g_4 \in Z_P^*$, such that the discrete logarithm based on each other element is not known and any subset $X \subset \chi(Z_P, g_1, g_2, g_3, g_4)$ with the polynomial order $|X|$, find a pair $(x_1, x_2, x_3, x_4) \in Z_P \times Z_q^3$, such that $x_1 g_1^{x_1} g_2^{x_2} = g_3^{x_3} g_4^{x_4} \bmod P$ and $(x_1, x_2, x_3, x_4) \notin X$.

Assumption (strong MDLP assumption). There is no probabilistic polynomial–time algorithm P that can solve the problem above.

Similar to Miyaji and Umeda [20], we have the following theorem to guarantee the security of the certificate-generation algorithm.

Theorem 1. *Let A be a polynomial–time adversary who can successfully forge a new certificate, then there exists an adversary B who can attack against the strong MDLP with at least the same advantage.*

4.3 Security Proof on the Group Signature

The following theorem shows the security of the group signature generation.

Theorem 2. *The interactive version of the group signature generation is an honest-verifier zero-knowledge proof of knowledge of a delegation key authorized by the* *original signer O and a pair of membership certificate and corresponding membership key obtained by interaction with the issuer. Furthermore,* (T_4, T_5) *is the ElGamal ciphertext for A with respect to the public key* y_{ok}.

Proof sketch. It is easy to prove the completeness and the zero-knowledgeness. The rest is to extract the knowledge. Owing to the knowledge extractor of each SPK, we can extract the values α_1, α_2, α_3, α_4, α_5, α_6, α_7 and α_8 satisfying

$$T_1 = \tilde{g}^{g_3^{\alpha_2}} \bmod \tilde{P} \tag{1}$$

$$T_2 = T_1^{g_4^{\alpha_1}} \bmod \tilde{P}, \tag{2}$$

$$T_6 = T_2^{g_3^{\alpha_3}} \bmod \tilde{P}, \tag{3}$$

$$T_3 = g_3^{\alpha_1} g_4^{\alpha_2} g_2^{\alpha_3} = g_3^{\alpha_5} g_4^{\alpha_7} g_2^{\alpha_6} \bmod P, \tag{4}$$

$$T_4 = y_{ik}^{-\alpha_4} g_1^{-\alpha_5} g_2^{\alpha_6} g_3^{\alpha_7} g^{\alpha_8} \bmod P, \tag{5}$$

$$T_5 = y_{ok}^{\alpha_7} \bmod P, \tag{6}$$

$$\tilde{g}^{T_4} = T_1^{\alpha_4} \bmod \tilde{P}, \tag{7}$$

$$T_o = t_o^{\alpha_8} \bmod P, \tag{8}$$

$$\alpha_4 \in Z_P. \tag{9}$$

Here, $\alpha_1 = \alpha_5$, $\alpha_2 = \alpha_7$ and $\alpha_3 = \alpha_6$ hold from Eq. (4). Thus, Eqs. (1) and (2) represent

$$T_1 = \tilde{g}^{g_3^{\alpha_7}} \bmod \tilde{P} \tag{10}$$

and

$$T_2 = T_1^{g_4^{\alpha_5}} \bmod \tilde{P}. \tag{11}$$

From Eqs. (5) and (10), Eq. (7) can be rewritten as

$$\tilde{g}^{y_{ik}^{-\alpha_4} g_1^{-\alpha_5} g_2^{\alpha_6} g_3^{\alpha_7} g_3^{\alpha_8}} = (\tilde{g}^{g_3^{\alpha_7}})^{\alpha_4} \bmod \tilde{P}.$$

Then

$$g_2^{\alpha_6} g^{\alpha_8} = \alpha_4 y_{ik}^{\alpha_4} g_1^{\alpha_5} \bmod P \tag{12}$$

holds.

It is obvious that the tuple $\{\alpha_6, \alpha_8, \alpha_4, \alpha_5\}$ implies a valid delegation key, a membership certificate, and the corresponding membership key.

From Eqs. (8) and (12), the original signer O's delegation key α_8, i.e., s_o, is extracted. That is to say, our signature proves the knowledge of the delegation key authorized by the original signer O.

From Eq. (12), Eq. (5) represents

$$T_4 = \alpha_4 g_3^{\alpha_7} \bmod P.$$

Thus, (T_4, T_5) is a pair of ElGamal ciphertext, and the ciphertext can be considered as the encryption of α_4, i.e., A, with respect to the opener's public key y_{ok}.

Besides, similar to Ref. [20], our proposed scheme satisfies the basic security requirements of the group signature scheme. We omit the proof procedures here. □

5 Efficiency

In this section, we simply analyze the performance of our proposed group–proxy signature-based e-cash system. Based on Miyaji and Umeda's [20] group signature scheme, the concept of proxy signature is drawn into. Moreover, *secure delegation* of proxy signature and *anonymity* of group signature are blended into each other. Here, we first define some notations in Table 1, then the time complexities of each phase of our scheme over Ref. [20] is presented in Table 2. From the comparison, only 9 modular multiplication operations and 17 modular exponentiation operations are increased additionally in one cycle of the procedure. This is much more efficient than "group signature + proxy signature" combined simply. Furthermore, the additional operations are negligible when the system is implemented with the large-scale integrated chip. More specifically, the additional operations over Miyaji and Umeda's [20] group signature scheme are as follows. In the *delegation*

Table 1. Definitions of notations.

Notation	Definition
T_{MUL}	Time for modular multiplication
T_{EXP}	Time for modular exponentiation

Table 2. Time complexity of the proposed scheme over Ref. [20].

Phase	T_{MUL}	T_{EXP}
Delegation key generation	2	4
Registration	1	1
Signature generation	3	6
Signature verification	3	6
Open	0	0

key generation phase, two modular multiplication operations and four modular exponentiation operations are respectively increased. In *registration* phase, one modular multiplication operation and one modular exponentiation operation are respectively increased. In the *signature generation* phase, three modular multiplication operations and six modular exponentiation operations are respectively increased. In the *signature verification* phase, three modular multiplication operations and six modular exponentiation operations are respectively increased. There is no additional operation increment in the *open* phase.

From Table 2, we can see that the time complexity of our proposed scheme is almost the same as Miyaji and Umeda's [20] original scheme, even though the group–proxy property is achieved creatively. Therefore, the implementation of our proposed e-cash system is applicable.

6 Conclusion

We have proposed the group–proxy signature model and presented a concrete scheme. Our scheme is an extension of the group signature scheme proposed by Miyaji and Umeda [20]. The basic group signature scheme is based on only publicly known-order groups. As a result, the ultimate group signature size and the computational amount of the signature generation and verification algorithms are reduced largely. Our security and performance analysis shows that the our proposed scheme is secure and applicable. We also demonstrate how our construction could be applied to set up an e-cash system in which more than one branch banks can be designated by the central bank to dispense anonymous e-cash on behalf of the central bank.

Acknowledgments: The authors would like to thank the anonymous referees and reviewers for their suggestions to improve this article. This research was supported by the National Natural Science Foundation of China under grant no. 61100214.

Received January 23, 2013; previously published online April 24, 2013.

Bibliography

[1] M. Abe, J. Groth, K. Haralambiev and M. Ohkubo, Optimal structure-preserving signatures in asymmetric bilinear groups, in: *CRYPTO 2011*, Lect. Notes Comput. Sci. 6841, pp. 649–666, Springer, Santa Barbara, California, USA, 2011.

[2] G. Ateniese, J. Camenisch, M. Joye and G. Tsudik, A practical and provably secure coalition-resistant group signature scheme, in: *Advances in Cryptology – Proceedings of CRYPTO2000*, M. Franklin and M. Bellare, eds., Lect. Notes Comput. Sci. 1880, pp. 255–270, Springer, Santa Barbara, California, USA, 2000.
[3] X. Boyen and B. Waters, Full-domain subgroup hiding and constant-size group signatures, in: *PKC 2007*, A. Yao, T. Okamoto and X. Wang, eds., Lect. Notes Comput. Sci. 4450 (2007), 1C15.
[4] E. Brickell, L. Chen and J. Li, A new direct anonymous attestation scheme from bilinear maps, in: *Trust 2008*, L. Chen and C. J. Mitchell, eds., Lect. Notes Comput. Sci. 4968 (2008), 1C20.
[5] J. Camenisch and M. Michels, A group signature scheme based on an RSAvariant. Preliminary version, in: *Advances in Cryptology – ASIACRYPT'98*, K. Feng, K. Ohta and D. Pei, eds., Tech. Rep. RS-98-27, BRICS, 1998.
[6] J. Camenisch and M. Stadler, Efficient group signatures for large groups, in: *Proceedings of CRYPTO '97*, B. Kaliski, ed., Lect. Notes Comput. Sci. 1294, pp. 410–424, Springer-Verlag, Santa Barbara, California, USA, 1997.
[7] J. Camenisch, A. Kiayias and M. Yung, On the portability of generalized schnorr proofs, in: *EUROCRYPT 2009*, A. Joux, ed., Lect. Notes Comput. Sci. 5479 (2010), 425C442.
[8] F. Cao and Z. F. Cao, A secure identity-based multi-proxy signature scheme, *Comput. Elect. Eng.* **35** (2009), 86–95.
[9] D. Chaum, Blind signatures for untraceable payments, in: *Advances in Cryptology, Proceedings of Crypto '82*, D. Chaum and R. Rivest, eds., pp. 199–203, Springer, Santa Barbara, California, USA, 1982.
[10] D. Chaum and E. van Heyst, Group signatures, in: *Proceedings of EUROCRYPT 91*, A. J. Clark and D. W. Davies, eds., Lect. Notes Comput. Sci. 547 (1991), 257–265.
[11] G. Fuchsbauer and D. Pointcheval, Anonymous proxy signatures, in: *SCN '08*, R. Ostrovsky, C. Blundo and X. Boyen, eds., Lect. Notes Comput. Sci. 5229 (2008), 201–217.
[12] E. Fujisaki, Sub-linear size traceable ring signatures without random oracles, in: *CT-RSA 11*, G. Ateniese and S. Boldyreva, eds., Lect. Notes Comput. Sci. 6558 (2011), 393–415.
[13] S. Hwang and C. Shi, A simple multi-proxy signature scheme, in: *Proceedings of the Tenth National Conference on Information Security*, D. Wang, ed., pp. 134–138.
[14] N. Lan and S. Rei, Efficient and probably secure trapdoor-free group signature schemes from bilinear pairings, in: *ASIACRYPT 2004*, K. Kim and P. J. Lee, eds., Lect. Notes Comput. Sci. 3329 (2004), 372–386. http://eprint.iacr.org/2004/104/ [full version].
[15] Z. H. Liu, Y. P. Hu and H. Ma, Secure proxy multi-signature scheme in the standard model, in: *Proceedings of the Second International Conference, ProvSec 2008,* J. Baek, et al., eds., Lect. Notes Comput. Sci. 5324 (2008), 124–140.
[16] Z. H. Liu, Y. P. Hu, X. S. Zhang and H. Ma, Secure proxy signature scheme with fast revocation in the standard model, *J. China Univ. Posts Telecomm.* **16** (2009), 116–124.
[17] A. Lysyanskaya and Z. Ramzan, Group blind digital signatures: a scalable solution to electronic cash, in: *Financial Cryptography (FC'98)*, R. Hirschfeld, ed., Lect. Notes Comput. Sci. 1465 (1998), 184–197.
[18] M. Mambo, K. Usuda and E. Okamoto, Proxy signatures for delegation signing operation, in: *Proceedings of the Third ACM Conference on Computer and Communication Security*, L. Gong, ed., pp. 48–57, 1996.
[19] M. Mambo, K. Usuda and E. Okamoto, Proxy signatures: delegation of the power to sign message, *IEICE Trans. Fundam.* **E79-A** (1996), 1338–1354.

[20] A. Miyaji and K. Umeda, A fully-functional group signature scheme over only known-order group, in: *Applied Cryptography and Network Security (ACNS 2004)*, M. Jakobsson and M. Yung, eds., Lect. Notes Comput. Sci. 3089 (2004), 164–179.
[21] D. Pointcheval and J. Stern, Security proofs for signature schemes, in: *Proceedings of EUROCRYPT 96*, J. Pastor and U. Maur, eds., Lect. Notes Comput. Sci. 1163 (1996), 387–405.
[22] J. C. Schdult, K. Matsuura and K. G. Paterson, Proxy signatures secure against proxy key exposure, in: *Public Key Cryptography C PKC 2008*, R. Cramer, ed., Lect. Notes Comput. Sci. 4939 (2008), 344–359.
[23] C. Schnorr, Efficient identification and signatures for smart cards, in: *Proceedings of CRYPTO 89*, G. Brassard, ed., Lect. Notes Comput. Sci. 435 (1990), 239–252.
[24] Y. Sun, C. Xu, Y. Yu and B. Yang, Improvement of a proxy multi-signature scheme without random oracles, *Comput. Commun*. **34** (2011), 257–263.
[25] G. K. Verma, A proxy signature scheme over braid groups, *Cryptology Eprint Archive Report*, 2008. http://www.eprint.iacr.org/2008/160.
[26] L. Yi, G. Bai and G. Xiao, Proxy multi-signature scheme, *Electron. Lett.* **36** (2000), 527–528.
[27] H. Zhou, B. Wang, L. Tie and J. Li, An electronic sash system with multiple banks based on proxy signature scheme, *J. Shanghai Jiao Tong Univ.* **38** (2004), 79–82 [in Chinese].