# A Novel Group Signature Scheme Based on MPKC

Guangdong Yang, Shaohua Tang *, and Li Yang

School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

**Abstract.** Group signature allows a group member to sign messages anonymously on the behalf of a group. In the case of a dispute, the designated group manager can open the signature to reveal the identity of its originator. As far as we know, most of the group signatures are based on traditional cryptography, such as RSA and discrete logarithm. Unfortunately these schemes would be broken if quantum computers emerge. The $\mathcal{MQ}$-problem based Multivariate Public-Key Cryptosystem (MPKC) is an important alternative to traditional PKCs for its potential to resist future attacks of quantum computers. The first group signature scheme based on MPKC is proposed in this paper. This scheme owns two special but important features. First, the group signature can be divided into different time periods. The signatures are linkable in the same time period, but un-linkable between different time periods. Second, the privileges of the group manager is limited. The group manager can not open a signature without the help of the verifier. These features are important in some applications such as e-voting systems. The theory of this scheme is simple and its security relies on the Isomorphism of Polynomials (IP) Problem and random hash function.

**Keywords:** multivariate public-key cryptosystem, group signature, isomorphism of polynomials, e-voting

## 1 Introduction

The concept of group signature was first introduced by Chaum and van Heyst [1] in 1991. A group signature scheme allows members of a group to sign messages on behalf of the group. A verifier can only tell that the signature is signed by a member of the group, but can not discover the identity of the signer. Furthermore, the verifier can't distinguish whether two signatures are issued by the same group member. However, in exceptional case such as legal dispute, the designated group manager can open a signature to reveal the identity of its originator. At the same time, no one(including the group manager) can forge a signature of other group members.

Because of its salient features, group signature attracts many researchers' attentions. Besides the first realization Chaum and van Heyst proposed in [1],

* Corresponding Author (email: shtang@IEEE.org)

there are many other schemes emerge in recent years. L. Chen and T. Pedersen proposed a new group signature scheme which hid the identity of the signer unconditionally and allowed new members to join the group [2]. J. Camenish presented a model of generalized group signature and its first realization [3]. This scheme allows the definition of coalition of group members to sign on the behalf of the group. In 1997, J. Camenish and M. Stadler proposed the first efficient group signature scheme for large groups, where the length of public keys and the length of signatures were independent of the number of group members [4]. Later G. Ateniese proposed a group signature scheme based on other number theory assumptions with better efficiency and stronger security [5]. More researches on group signature include group signatures for hierarchical multi-groups [6], group blind signatures [7], multi-group signatures [8], sub-group signatures [8], ID-based group signatures [9], and so on.

The properties of group signature make it attractive for many specialized applications, such as e-bidding, e-voting, e-cash, and so on. For example, in e-bidding systems all companies with invitation to tender form a group and each company uses the group signature to sign its tender anonymously. The verifier can check whether a tender was submitted by a company in the group, but he can't know which company submitted it. Once the preferred tender is selected, the winner can be traced while other bidders remain anonymous.

But in e-voting systems, the situation is a little bit different. The voters are not allowed to vote multiple times. So the tallying authority must be able to distinguish the reduplicative votes without opening the ballots. Moreover, there usually exists supervision authority to limit the privileges of the tallying authority and guarantee the justice of the voting in a voting system. Therefore, the general group signature schemes can not be applied to e-voting systems directly.

As far as we know, most of the group signatures are based on traditional cryptography, such as RSA and discrete logarithm. Unfortunately, the algorithm discovered by Peter Shor shows that finding discrete logarithms and factoring integers can be accomplished in polynomial time on a quantum computer [10]. When the quantum computers arrive, the traditional PKCs based on these problem, such as RSA and ECC, will be broken. Multivariate public-key cryptosystem (MPKC) is considered to be one of the best alternative. The security foundation of MPKC is the knowledge that solving a set of multivariate polynomial equations over a finite filed is a NP-hard problem [11]. Quantum computers do not appear to have any advantage when dealing with this NP-hard problems, and it seems that we can not find a solution to a set of polynomial equations efficiently even in the future. Moreover, MPKC schemes are more efficient than traditional PKCs. It makes them suitable for limited computing devices, such as smart cards. Various MPKC schemes have been proposed in recent years, for example, [12], [13], [14], [15], [16].

In most MPKC schemes, the public key is a set of polynomials $p = (p_1, \cdots, p_m)$ in variables $x = (x_1, \cdots, x_n)$, where all variables and coefficients are in $K =$

$GF(k)$. This is usually accomplished via $\mathcal{P} = S \circ \mathcal{Q} \circ T : K^n \to K^m$, or

$$\mathcal{P} : w = (w_1, \cdots, w_n) \in K^n \overset{T}{\mapsto} x = M_T w + c_T$$
$$\overset{\mathcal{Q}}{\mapsto} y \overset{S}{\mapsto} z = M_S y + c_S = (z_1, \cdots, z_m) \in K^m$$

The central map $\mathcal{Q}$ belongs to a certain class of quadratic maps whose inverse can be computed easily. The maps $S$ and $T$ are affine and bijective. The private key consist of the central map $\mathcal{Q}$ and affine maps $S$, $T$. For more details about MPKC, we refer the reader to [17].

*Our Contribution:* This paper proposes the first group signature scheme based on multivariate public-key cryptosystem. Compared with general group signature schemes, our scheme is specific to e-voting systems. Our scheme can satisfy the requirements of e-voting systems we mention above due to two special properties, *Special Unlinkability* and *Special Traceability*. *Special Unlinkability* means that the group signature can be divided into several time periods. The signatures are linkable in the same time period, but un-linkable between different time periods. So, the tallying authority can detect reduplicative votes before *OPEN* them. *Special Traceability* means that the group manager can not open the signatures alone. In order to reveal the identity of the signer, he has to cooperate with the verifier. So there are two managements in our group signature, which is common in voting systems for the tallying authority and supervision authority. Besides, the length of signatures and the computation overhead are independent of the number of group members in our scheme. Therefore, it's efficient for large groups.

The rest of this paper is organized as follows. The next section presents the formal model of our group signature scheme. Section 3 introduces the signature of knowledge protocol based on MPKC, which is the basic building block of our scheme. The details of our novel group signature scheme are described in section 4. The security of our scheme is discussed in section 5. Our scheme is compared with other schemes in section 6. Finally, the paper concludes in section 7.

## 2 The Model of Our Group Signature Scheme

The traditional group signature schemes allow a member to sign messages anonymously on the behalf of his group. The verifier can verify that the signature was signed by a member of the group, but can not discover which group member made it. Moreover, the signatures are un-linkable, which means, given two group signatures, the verifier can't know whether they are issued by the same group member or not. But the *unlinkability* seems not adaptive to some applications such as e-voting systems. The verifier need to be able to distinguish the reduplicative votes, meanwhile the anonymity should be guaranteed. Besides, in traditional group signatures, the group manager can open any group signature arbitrarily. But in the e-voting systems, there usually exists a supervision authority to guarantee the justice of the voting. Therefore we need to limit the privileges of

the group manager. So, the traditional group signatures can not be applied to e-voting systems directly.

In order to solve these problems, we propose a novel group signature scheme specific to e-voting systems. Our scheme is defined as follows.

- *Correctness:* A group signature issued by the valid group member must be accepted by the verifier.
- *Unforgeability:* Only the valid group members are able to sign messages on the behalf of the group.
- *Exculpability:* Neither the group member nor the group manager is able to sign messages on the behalf of other group members. This also makes the group signature undeniable.
- *Anonymity:* Given a valid group signature, no one can reveal the identity of the actual signer, unless the group manager opens the signature with the help of the verifier.
- *Special Unlinkability:* The group signature can be divided into several time periods. The signatures in the same time period are linkable. The verifier can verify whether two signatures are issued by the same group member. But the signatures between different time periods are un-linkable. They are completely anonymous.
  This feature is useful in e-voting systems. For example, in an election conference, there are several positions need to be selected. Thus the voting can be divided into several time periods, each period for one position. As the votes in the same period are linkable, the verifier is able to distinguish the reduplicative votes without opening them. At the same time, the votes between different periods are un-linkable, so the votes for different position are anonymous.
- *Special Traceability:* Neither the group manager nor the verifier can reveal the identity of a signature alone, but they can do it together.
  In our scheme, the group manager can not open a signature arbitrarily. His privileges is limited. This property is important in some application such as e-voting systems. The management usually consists of the tallying authority and the supervision authority. Neither of them can reveal the identity of a ballot alone. They have to work together to open a ballot to reveal the actual identity. This property makes the voting more open.

## 3 Signature of Knowledge Based on MPKC

In this section we introduce the signature of knowledge protocol based on MPKC, which is the building block of our scheme.

### 3.1 Isomorphism of Polynomials (IP) Problem

The Isomorphism of Polynomials (IP) problem was first introduced by Patarin in [13]. It is a fundamental problem of multivariate cryptography as it is related

to the hardness of the key recovery of such cryptosystems. The concept of IP is described as follows. For more details, we refer the reader to [13].

Let $K = GF(k)$ be a finite field. All variables are over the field $K$.

Let $A$ be a set of $u$ quadratic equations with $n$ variables $x_1, \cdots, x_n$ that give the $y$ values from the $x$ values:

$$y_k = \sum_i \sum_j \gamma_{ijk} x_i x_j + \sum_i \mu_{ik} x_i + \delta_k \quad , \quad k = 1, \cdots, u$$

Let $B$ be a set of $u$ quadratic equations with $n$ variables $x'_1, \cdots, x'_n$ that give the $y'$ values from the $x'$ values:

$$y'_k = \sum_i \sum_j \gamma'_{ijk} x'_i x'_j + \sum_i \mu'_{ik} x'_i + \delta'_k \quad , \quad k = 1, \cdots, u$$

Let $T$ be a bijective affine transformation of the variables $x'_i$, $1 \leq i \leq n$, $S$ be a bijective affine transformation of the variables $y_k$, $1 \leq k \leq u$. That is :

$$T(x'_1, \cdots, x'_n) = (x_1, \cdots, x_n), S(y_1, \cdots, y_u) = (y'_1, \cdots, y'_u)$$

*If there exists such transformation pair $(S, T)$ and satisfy $B = S \circ A \circ T$, Then we call $A$ and $B$ are "isomorphic", and the bijective affine transformation pair $(S, T)$ is an "isomorphism" from $A$ to $B$.*

*The Isomorphism of Polynomials Problem is : if $A$ and $B$ are two public sets of $u$ quadratic equations, and $A$ and $B$ are isomorphic, find an isomorphism $(S, T)$ from $A$ to $B$.*

### 3.2 Signature of Knowledge Protocol

Zero-knowledge Proofs of Knowledge allow a prover to demonstrate the knowledge of a secret, but without leaking any useful information.

Patarin presented a Non-interactive Zero-knowledge Proofs of knowledge scheme based on the Isomorphism of Polynomials Problem in [13]. In this paper, we will use a simpler but workable scheme as follows. We call it signature of knowledge.

Parameters: Let $K$ be a finite field. Let $n$, $u$, $q$ be three integers. $\mathcal{H}$ is a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^q$ which maps a binary string of arbitrary length to a $q$-bit hash value.

**Definition 1.** *Let $A$ and $B$ be two sets of $u$ quadratic equations with $n$ variables as follows:*

$$A : y_k = \sum_i \sum_j \gamma_{ijk} x_i x_j + \sum_i \mu_{ik} x_i + \delta_k \quad , \quad k = 1, \cdots, u$$

$$B : y'_k = \sum_i \sum_j \gamma'_{ijk} x'_i x'_j + \sum_i \mu'_{ik} x'_i + \delta'_k \quad , \quad k = 1, \cdots, u$$

*Let $m$ be the message to be signed. We call the $q+1$ tuples $V = [H, (S_1, T_1), (S_2, T_2), \ldots, (S_q, T_q)]$ a signature of knowledge related to $(A, B)$ and message $m$, if and only if the following conditions are satisfied:*

$$C_i = \begin{cases} S_i \circ A \circ T_i, & if \ \ H(i) = 0 \\ S_i \circ B \circ T_i, & if \ \ H(i) = 1 \end{cases}, \qquad 1 \leq i \leq q$$

*and* $H = \mathcal{H}(m||C_1||C_2||\ldots||C_q)$.

If the prover knows the isomorphism $(S, T)$ from $A$ to $B$, that means $B = S \circ A \circ T$, then the signature of knowledge can be constructed as follows.

First, the prover selects $q$ random bijective affine transformation pairs $(S'_1, T'_1), (S'_2, T'_2), \cdots, (S'_q, T'_q)$, which look like:

$$\begin{array}{l} T'_1(x_1^{(1)}, \cdots, x_n^{(1)}) = (x_1, \cdots, x_n), \ \ S'_1(y_1, \cdots, y_u) = (y_1^{(1)}, \cdots, y_u^{(1)}) \\ \cdots\cdots \\ T'_q(x_1^{(q)}, \cdots, x_n^{(q)}) = (x_1, \cdots, x_n), \ \ S'_q(y_1, \cdots, y_u) = (y_1^{(q)}, \cdots, y_u^{(q)}) \end{array}$$

The prover computes

$$\begin{array}{l} C_1 = S'_1 \circ A \circ T'_1 \\ C_2 = S'_2 \circ A \circ T'_2 \\ \cdots\cdots \\ C_q = S'_q \circ A \circ T'_q \end{array}$$

And obtains $q$ sets of $u$ quadratic equations with $n$ variables as follows:

$$\begin{array}{l} y_k^{(1)} = \sum\limits_i \sum\limits_j \gamma_{ijk}^{(1)} x_i^{(1)} x_j^{(1)} + \sum\limits_i \mu_{ik}^{(1)} x_i^{(1)} + \delta_k^{(1)} \quad , \ \ k = 1, \cdots, u \\ \cdots\cdots \\ y_k^{(q)} = \sum\limits_i \sum\limits_j \gamma_{ijk}^{(q)} x_i^{(q)} x_j^{(q)} + \sum\limits_i \mu_{ik}^{(q)} x_i^{(q)} + \delta_k^{(q)} \quad , \ \ k = 1, \cdots, u \end{array}$$

Second, the prover computes the hash value $H = \mathcal{H}(m\|C_1\|C_2\|\ldots\|C_q)$, where $\|$ is the concatenation function. Suppose the binary format of value $H$ is presented as $H(q)\ldots H(2)H(1) \in \{0,1\}^q$. The prover computes

$$(S_i, T_i) = \begin{cases} (S'_i, T'_i), & if \ H(i) = 0 \\ (S'_i \circ S^{-1}, T^{-1} \circ T'_i), & if \ H(i) = 1 \end{cases}, \qquad 1 \leq i \leq q$$

Thus, the signature of knowledge $V$ is constructed.

When the verifier receives the public key $(A, B)$, the message $m$ , and the signature of knowledge $V$, the verification of the signature is easy.

First, the verifier computes $q$ sets of $u$ quadratic equations via

$$C_i = \begin{cases} S_i \circ A \circ T_i, & if \ H(i) = 0 \\ S_i \circ B \circ T_i, & if \ H(i) = 1 \end{cases} \qquad 1 \leq i \leq q$$

Then the verifier computes the hash value $H' = \mathcal{H}(m\|C_1\|C_2\|\ldots\|C_q)$, and checks whether the new hash value $H'$ is equal to $H$ .

## 4 A Novel Group Signature Scheme Based on MPKC

In this section we present a novel group signature scheme based on MPKC, specific to e-voting systems.

### 4.1 System Setup

The group manager setup the system's parameters.

- Let $n, u, q$ be three integers. $K = GF(k)$ be a finite field.
- $\mathcal{H}$ is a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^q$ which maps a binary string of arbitrary length to a $q$-bit hash value.
- $\mathcal{Q}$ is the central map selected by the group mannager.
- Each user $i$ has his private key $(L_{1,i}, L_{2,i})$, and his public key $A_i = L_{1,i} \circ \mathcal{Q} \circ L_{2,i}$.

### 4.2 Joining Group

Every user needs to join the group to obtain the qualification of signing on the behalf of the group. Suppose a user $i$ wants to join the group.

*JOIN:*

1. User $i$ selects a random bijective affine transformation pair $(S_i, T_i)$, and maps $A_i$ to $B_i$ via $B_i = S_i \circ A_i \circ T_i$. Thus, $A_i$ and $B_i$ are isomorphic, and the affine transformation pair $(S_i, T_i)$ is the isomorphism from $A_i$ to $B_i$. We call $(A_i, B_i)$ the public key of user $i$ in the group, and call $(S_i, T_i)$ the private key of user $i$ in the group.
2. User $i$ generates a signature of knowledge of $(A_i, B_i)$, denoted by $V_i$, and sends the knowledge $V_i$ and $(A_i, B_i)$ to the group manager. We assume the communication between the user $i$ and the group manager is secure, i.e., private and authentic.
3. The group manager verifies the signature of knowledge. If it's valid, he saves the knowledge $V_i$ and $(A_i, B_i)$. Otherwise, he deletes the knowledge $V_i$ and $(A_i, B_i)$ and refuses user $i$'s application.

### 4.3 Signing Messages

The group manager can divide the group signature into several time periods. The change of the period is controlled by the group manager.

In order to change a period, the group manager executes the following steps.

*CHANGE PERIOD:*

1. The group manager selects the public keys of some group members. These members form a sign group.
2. For every public key $(A_i, B_i)$ in the sign group, the group manager generates a random bijective affine transformation pair $(S_{1,i}, T_{1,i})$, and computes

$$A_i^{(1)} = S_{1,i} \circ A_i \circ T_{1,i}$$
$$B_i^{(1)} = S_{1,i} \circ B_i \circ T_{1,i}$$

The group manager publishes $(A_i^{(1)}, B_i^{(1)})$ to the bulletin board. Then he sends every affine transformation pair $(S_{1,i}, T_{1,i})$ to the corresponding member $i$ in the sign group via the secure channel.

Within each period, group members in the sign group can sign any messages on the behalf of the group. But the signatures are linkable. Suppose group member $i$ wants to sign a message $m$, he will execute the following steps.

*SIGN:*

1. The member $i$ sends his $(A_i^{(1)}, B_i^{(1)})$ to verifier $v$.
2. The verifier $v$ checks whether $(A_i^{(1)}, B_i^{(1)})$ is in the bulletin board. If yes, he selects a random bijective affine transformation pair $(S_{2,i}, T_{2,i})$ and sends them to member $i$. Then he computes $A_i^{(2)} = S_{2,i} \circ A_i^{(1)} \circ T_{2,i}$ , $B_i^{(2)} = S_{2,i} \circ B_i^{(1)} \circ T_{2,i}$ , and saves $(A_i^{(2)}, B_i^{(2)})$ and $(S_{2,i}, T_{2,i})$ in his database.
3. The member $i$ computes $(A_i^{(2)}, B_i^{(2)})$ as the verifier did. Meanwhile he computes the isomorphism $(S_i', T_i')$ from $A_i^{(2)}$ to $B_i^{(2)}$ via

$$S_i' = (S_{2,i} \circ S_{1,i}) \circ S_i \circ (S_{2,i} \circ S_{1,i})^{-1}$$
$$T_i' = (T_{1,i} \circ T_{2,i})^{-1} \circ T_i \circ (T_{1,i} \circ T_{2,i})$$

With $(S_i', T_i')$, member $i$ can generate the signature of knowledge of message $m$ and $(A_i^{(2)}, B_i^{(2)})$, denoted by $V_i^{(2)}$. Thus $[V_i^{(2)}, (A_i^{(2)}, B_i^{(2)})]$ is the group signature issued by member $i$.

### 4.4 Verifying Signatures

After receiving the group signature $[V_i^{(2)}, (A_i^{(2)}, B_i^{(2)})]$, the verifier $v$ can check the legality of the signature via two steps.

*VERIFY:*

1. The verifier $v$ recovers the $(A_i^{(1)}, B_i^{(1)})$ via

$$A_i^{(1)} = S_{2,i}^{-1} \circ A_i^{(2)} \circ T_{2,i}^{-1}$$

$$B_i^{(1)} = S_{2,i}^{-1} \circ B_i^{(2)} \circ T_{2,i}^{-1}$$

Then he checks whether $(A_i^{(1)}, B_i^{(1)})$ exists in the bulletin board (This operation can be finished in constant time via some special technologies, such as hash table.). If yes, he goes to next step. Otherwise the signature is illegal.

2. The verifier $v$ checks whether $V_i^{(2)}$ is a valid signature of knowledge of message $m$ and $(A_i^{(2)}, B_i^{(2)})$ via the steps in 3.2. If yes, he accepts the signature. Otherwise the signature is illegal.

### 4.5 Linking Signatures

Given two group signatures $[V_i^{(2)}, (A_i^{(2)}, B_i^{(2)})]$ and $[V_j^{(2)}, (A_j^{(2)}, B_j^{(2)})]$, the judgment of reduplicative signatures is easy.

*LINK:*

1. The verifier checks whether $(A_i^{(2)}, B_i^{(2)})$ is equals to $(A_j^{(2)}, B_j^{(2)})$. If yes, the two signatures are issued by the same member. Otherwise, goto next step.
2. The verifier recovers the $(A_i^{(1)}, B_i^{(1)})$ and the $(A_j^{(1)}, B_j^{(1)})$ as follows:

$$(A_i^{(1)}, B_i^{(1)}) = (S_{2,i}^{-1} \circ A_i^{(2)} \circ T_{2,i}^{-1}, S_{2,i}^{-1} \circ B_i^{(2)} \circ T_{2,i}^{-1})$$
$$(A_j^{(1)}, B_j^{(1)}) = (S_{2,j}^{-1} \circ A_j^{(2)} \circ T_{2,j}^{-1}, S_{2,j}^{-1} \circ B_j^{(2)} \circ T_{2,j}^{-1})$$

If $(A_i^{(1)}, B_i^{(1)})$ and $(A_j^{(1)}, B_j^{(1)})$ are still different, then the two signatures are issued by different group members. Otherwise, they are issued by the same member.

Note that if two signatures are from different periods, they are un-linkable. That means that the result of the *LINK* operation is always *"issued by different group members"*.

### 4.6 Opening Signatures

In order to open a group signature to reveal the actual identity of the signer, the group manager has to cooperate with the verifier. For example, to open the group signature $[V_i^{(2)}, (A_i^{(2)}, B_i^{(2)})]$, the group manager receives an affine transformation pair $(S_{2,i}, T_{2,i})$ from the verifier. Then the group manager computes $(A_i, B_i)$ using $(S_{2,i}, T_{2,i})$ together with his own $(S_{1,i}, T_{1,i})$:

$$A_i = (S_{2,i} \circ S_{1,i})^{-1} \circ A_i^{(2)} \circ (T_{1,i} \circ T_{2,i})^{-1}$$
$$B_i = (S_{2,i} \circ S_{1,i})^{-1} \circ B_i^{(2)} \circ (T_{1,i} \circ T_{2,i})^{-1}$$

The isomorphism from $(A_i^{(2)}, B_i^{(2)})$ to $(A_i, B_i)$ is $((S_{2,i} \circ S_{1,i})^{-1}, (T_{1,i} \circ T_{2,i})^{-1})$. Thus the group manager gets the identity of the actual signer.

## 5 Security Analysis

### 5.1 Security of IP Problem

Due to its nice properties, the Isomorphism of Polynomials has been used in numerous applications, such as the authentication / signature schemes proposed by J. Patarin in [13] and the traitor tracing scheme described by O. Billet and H. Gilbert in [18]. At the same time, many researchers are trying to solve the IP problem in an efficient way.

N. Courtois proposed the first significant algorithm for IP in [19], known as the "To and Fro" technique. This algorithm assumes the ability to invert the polynomial systems, therefore has an exponential complexity. Moreover, this paper gave an upper bound on the theoretical complexity of IP. Perret and Faugère pointed out that the proof in [19] was not complete. They gave an upper bound on the theoretical complexity of "IP-like" problems, and presented a new algorithm for solving IP when S and T are linear mappings [20]. C. Bouillaguet presented an improved algorithm in [21]. Their algorithm combined several new techniques and they claimed that their algorithm got the best result on the state of the art.

An important special case of IP is the IP problem with one secret (IP1S for short). Although most of the algorithms for IP can be applied to IP1S almost directly, several algorithms are proposed to solve IP1S. Geiselmann proposed the first algorithm for IP1S, using an exhaustive search to find the solutions of an algebraic system of equations [22]. Later, Levy-dit-Vehel and Perret improved it by using the Gröbner basis computation [23]. Perret presented a new approach for solving IP1S using the Jacobian matrix [24]. The algorithm is polynomial when the number of polynomials $u$ is equal to the number of variables $n$, but inefficient when $u$ is much smaller than $n$.

The best algorithms for IP and IP1S, as far as we know, are summarized in Table 1. We can see that the best algorithm for some instances of IP is still exponential. Solving the IP problem is computationally hard if we choose the parameter properly.

To strengthen the security of our scheme, we suggest that the parameters of our scheme should satisfy the following conditions:
1) the transformations should be affine;
2) the polynomials in $\mathcal{Q}$ should be homogeneous;
3) the number of polynomials $u$ should be smaller than the number of variables $n$.

### 5.2 Security of Our Scheme

1. *Correctness:* A group signature issued by the valid group member must be accepted by the verifier.

**Table 1.** The Best Algorithms for IP and IP1S

| Problem | Subcase | Complexity |
|---|---|---|
| IP1S | degree = 2 | $\mathcal{O}(n^6)$ |
| | degree = 3, $u \ll n$, inhomogeneous | $\mathcal{O}(n^6)$ |
| | degree = 3, $u \ll n$, homogeneous | $\mathcal{O}(n^6 \cdot q^{n/2})$ |
| IP | degree = 2, $u = n$, inhomogeneous | Heuristic:$\mathcal{O}(n^3)$ / Rigorous:$\mathcal{O}(n^6)$ |
| | degree = 2, $u = n$, homogeneous | $\mathcal{O}(n^{3.5} \cdot q^{n/2})$ |

*Notation: $u$ is the number of polynomials in $\mathcal{Q}$, $n$ is the number of variables.*

*Proof.* By inspection.

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

   *Proof.* As mentioned in section 4.3, the verifier will check the public keys in the bulletin board. So in order to get the transformation pair $(S_{2,i}, T_{2,i})$ from verifier and generate a signature, the user $i$ has to get the qualification from the group manager.

3. *Anonymity:* Given a valid group signature, no one can reveal the identity of the actual signer, unless the group manager opens the signature with the help of the verifier.

   *Proof.* Given a group signature $[V_i^{(2)}, (A_i^{(2)}, B_i^{(2)})]$, in order to reveal the identity of the signer, the attacker has to find the affine transformation pair $(S_i', T_i')$ that maps $(A_i, B_i)$ to $(A_i^{(2)}, B_i^{(2)})$. That means that he has to solve the *Isomorphism of Polynomials Problem*. As we seen in section 5.1, it is computationally hard without $(S_{1,i}, T_{1,i})$ and $(S_{2,i}, T_{2,i})$, which owned by the group manager and the verifier respectively.

4. *Exculpability:* Neither the group member nor the group manager is able to sign messages on the behalf of other group members.

   *Proof.* Suppose the group member $j$ wants to sign messages on the behalf of the group member $i$. Member $j$ may ask the group manager for the affine transformation pair $(S_1, T_1)$ and ask the verifier for the affine transformation pair $(S_2, T_2)$. Then he can compute $(A_i^{(2)}, B_i^{(2)})$:

   $$A_i^{(2)} = S_2 \circ S_1 \circ A_i \circ T_1 \circ T_2$$
   $$B_i^{(2)} = S_2 \circ S_1 \circ B_i \circ T_1 \circ T_2$$

   In order to generate a signature of knowledge of $(A_i^{(2)}, B_i^{(2)})$, he need to find out the isomorphism from $A_i^{(2)}$ to $B_i^{(2)}$. That means that he has to solve the *Isomorphism of Polynomials Problem*. It is computationally hard without the private key $(S_i, T_i)$ of member $i$.

5. *Special Unlinkability*: The signatures are linkable in the same time period, but un-linkable in different time periods.

   *Proof.* In one period, group member $j$ can only get one "license" $(S_1, T_1)$ from the group manager. Thus he can only produce one $(A_j^{(1)}, B_j^{(1)})$. Maybe he could ask the verifier twice for $(S_2, T_2)$ and $(S_2', T_2')$, then produces $(A_j^{(2)}, B_j^{(2)})$ and $(A_j'^{(2)}, B_j'^{(2)})$, but the verifier will recover the same $(A_j^{(1)}, B_j^{(1)})$ using $(S_2, T_2)$ and $(S_2', T_2')$ respectively. So the signatures in the same time period are linkable.
   While in different periods, the member $j$ will get different "licenses" $(S_1, T_1)$ and $(S_1', T_1')$ from group manager, and he produces $(A_j^{(1)}, B_j^{(1)})$ and $(A_j'^{(1)}, B_j'^{(1)})$. The verifier may recover them using $(S_2, T_2)$ and $(S_2', T_2')$ , but he can not recover the origin $(A_j, B_j)$. Therefore, the signatures between different time periods are unlinkable.

6. *Special Traceability*: Neither the group manager nor the verifier can reveal the identity of a signature alone, but they can do it together.

   *Proof.* In our scheme, after the computation of $(A_i^{(1)}, B_i^{(1)})$, the signer $i$ maps his $(A_i^{(1)}, B_i^{(1)})$ to $(A_i^{(2)}, B_i^{(2)})$ by applying the affine transformation pair $(S_{2,i}, T_{2,i})$ given by the verifier. Thus, the group manager can not establishes the relationship between $(A_i^{(2)}, B_i^{(2)})$ and $(A_i^{(1)}, B_i^{(1)})$ without $(S_{2,i}, T_{2,i})$. So the group manager can not reveal the identity of a signature by himself. Similarly, the verifier can not establishes the relationship between $(A_i^{(1)}, B_i^{(1)})$ and original $(A_i, B_i)$. So the verifier can not open a signature alone either. However, as shown in 4.6, the group manager and the verifier can discover the identity of a signature if they work together.

## 6 Comparison

In Table 2, we compare our scheme with several classical group signature schemes in [1], [25], [5]. As far as we know, our scheme is the first group signature scheme based on MPKC, and is likely the first group signature scheme that resists the future attacks of quantum computers. So we can only compare our scheme with the schemes based on traditional cryptography.

The early proposed group signature schemes are inefficient for large groups, as the length of signatures or the computation of operations (i.e. *SIGN*, *VERIFY*, *OPEN*) is linear to the number of group members. Some practical group signature schemes for large groups are proposed in [4], [5]. In our scheme, the computation of all operations ( including *JOIN*, *SIGN*, *LINK*, *VERIFY* and *OPEN*) are independent of the number of group members. The length of signatures and the length of keys of users are constant. But the length of keys of group manager are linear in the number of group members, so reducing the storage of the scheme may be part of our future work. Nevertheless, this scheme is efficient for large groups.

Besides, compared with other schemes, our scheme can be applied to e-voting systems directly due to two special properties, the *special linkability* and the *special traceability*.

Table 2: Comparisons of Different Group Signatures

| Property | Cham in [1] | Petersen in [25] | Ateniese in [5] | Our Scheme |
|---|---|---|---|---|
| Cryptographic Assumption | Discrete Logarithm | Discrete Logarithm | Strong RSA and Diffie-Hellman | Isomorphism of Polynomials |
| Quantum Attack | yes | yes | yes | no |
| *SIGN* | $\mathcal{O}(n)$ | $\mathcal{O}(n)$ | constant | constant |
| *VERIFY* | $\mathcal{O}(n)$ | $\mathcal{O}(n)$ | constant | constant |
| *OPEN* | $\mathcal{O}(n)$ | $\mathcal{O}(n)$ | constant | constant |
| Length of Signature | $\mathcal{O}(n)$ | $\mathcal{O}(n)$ | constant | constant |
| Length of Key | $\mathcal{O}(n)$ | constant | constant | Manager:$\mathcal{O}(n)$ Users:constant |

*Notation: $n$ is the number of group members.*

## 7 Conclusion

This paper proposes the first group signature scheme based on multivariate public-key cryptosystem. Its security relies on the Isomorphism of Polynomials (IP) Problem in MPKC and random hash function. Compared with general group signature schemes, our scheme is specific to e-voting systems. Due to two special properties, *Special Unlinkability* and *Special Traceability*, our scheme can satisfy the extra requirements of e-voting systems. Besides, the length of signatures and the computation overhead are independent of the number of group members. Therefore, it's efficient for large groups.

## Acknowledgement

This paper is supported by the Fundamental Research Funds for the Central Universities of China under Grant No. 2009ZZ0035 and the Natural Science Foundation of Guangdong Province of China under Grant No. 9351064101000003.

## References

1. D. Chaum and E. van Heijst, "Group signatures," in *Advances in Cryptology-EUROCRYPT 1991*, vol. 547. Springer-Verlag, 1991, pp. 257–265.

2. L. Chen and T. P. Pedersen, "New group signature schemes," in *Advances in Cryptology-EUROCRYPT 1994*, vol. 950. Springer-Verlag, 1995, pp. 171–181.
3. J. Camenisch, "Efficient and generalized group signatures," in *Advances in Cryptology-EUROCRYPT 1997*, vol. 1233. Springer-Verlag, 1997, pp. 465–479.
4. J. Camenisch and M. Stadler, "Efficient group signatures schemes for large groups," in *Advances in Cryptology-CRYPTO 1997*, vol. 1296. Springer-Verlag, 1997, pp. 410–424.
5. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," in *Advances in Cryptology-CRYPTO 2000*, vol. 1880. Springer-Verlag, 2000, pp. 255–270.
6. S. Kim, S. Park, and D. Won, "Group signatures for hierarchical multigroups," in *Information Security*, vol. 1396. Springer-Verlag, 1998, pp. 273–281.
7. A. Lysyanskaya and Z. Ramzan, "Group blind digital signatures: A scalable solution to electronic cash," in *Financial Cryptography*, vol. 1465. Springer-Verlag, 1998, pp. 184–197.
8. G. Ateniese and G. Tsudik, "Some open issues and directions in group signatures," in *Proceedings of Financial Cryptography 1999*, vol. 1648. Springer-Verlag, 1999, pp. 196–211.
9. Y. Tseng and J. Jan, "A novel ID-based group signature," in *International computersymposium, workshop on cryptology and information security*, 1998, pp. 159–164.
10. P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35nd Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, 1994, pp. 124–134.
11. M. R. Garey and D. S. Johnson, *Computers and intractability. A guide to the theory of NP-completeness*. W.H. Freeman, 1979.
12. T. Matsumoto and H. Imai, "Public quadratic polynomial-tuples for efficient signature verification and message encryption," in *Advances in Cryptology-EUROCRYPT 1988*, vol. 330. Springer-Verlag, 1988, pp. 419–453.
13. J. Patarin, "Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms," in *Advances in Cryptology-EUROCRYPT 1996*, vol. 1070. Springer-Verlag, 1996, pp. 33–48.
14. J. Ding, "A new variant of the Matsumoto-Imai cryptosystem through perturbation," in *Public Key Cryptography-PKC 2004*, vol. 2947. Springer-Verlag, 2004, pp. 305–318.
15. B.-Y. Yang and J.-M. Chen, "A more secure and efficacious TTS signature scheme," in *Information Security and Cryptology-ICISC 2003*, vol. 2971. Springer-Verlag, 2003, pp. 320–338.
16. J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *Applied Cryptography and Network Security*, vol. 3531. Springer-Verlag, 2005, pp. 164–175.
17. J. Ding, J. E. Gower, and D. S. Schmidt, *Multivariate public key cryptosystems*. Springer-Verlag, 2006.
18. O. Billet and H. Gilbert, "A traceable block cipher," in *Advances in Cryptology-ASIACRYPT 2003*, vol. 2894. Springer-Verlag, 2003, pp. 331–346.
19. N. Courtois, L. Goubin, and J. Patarin, "Improved algorithms for isomorphisms of polynomials," in *Advances in Cryptology-EUROCRYPT 1998*, vol. 1403. Springer-Verlag, 1998, pp. 184–200.
20. J.-C. Faugère and L. Perret, "Polynomial equivalence problems: Algorithmic and theoretical aspects," in *Advances in Cryptology-Eurocrypt 2006*, vol. 4004. Springer-Verlag, 2006, pp. 30–47.

21. C. Bouillaguet, J.-C. Faugère, P.-A. Fouque, and L. Perret, "Differential-algebraic algorithms for the isomorphism of polynomials problem," http://eprint.iacr.org/2009/583.
22. W. Geiselmann and W. Meier, "An attack on the isomorphisms of polynomials problem with one secret," in *International Journal of Information Security*, vol. 2. Springer-Verlag, 2003, pp. 59–64.
23. F. L. dit Vehel and L. Perret, "Polynomial equivalence problems and applications to multivariate cryptosystems," in *Progress in Cryptology-INDOCRYPT 2003*, vol. 2904. Springer-Verlag, 2003, pp. 235–251.
24. L. Perret, "A fast cryptanalysis of the isomorphism of polynomials with one secret problem," in *Advances in Cryptology-EUROCRYPT 2005*, vol. 3494. Springer-Verlag, 2005, pp. 354–370.
25. H. Petersen, "How to convert any digital signature scheme into a group signature scheme," in *Security Protocols*. Springer-Verlag, 1998, pp. 177–190.