An Effective Method to Implement Group Signature with Revocation

He Ge

Department of Computer Science and Engineering
University of North Texas
Denton, TX 76203
Email: ge@unt.edu

Abstract. In this paper we propose an effective method to integrate revocation mechanism into some group signature schemes based on the strong RSA assumption. In such mechanism, the group manager can either update group member's certificates, or revoke a group member. More specifically, we propose a generic method for protocols of sign, verify, and revocation. We use an example to demonstrate the effectiveness of the method by applying it to a well known group signature scheme. The new construction has better performance while enjoying an efficient revocation mechanism.

Keywords: group signature, strong RSA assumption, anonymity, full anonymity, revocation.

1 Introduction

Group signature is a privacy-preserving signature scheme, which was introduced by Chaum and Heyst in 1991 [16]. In such scheme, a group member can sign a message on behalf of the group without revealing his identity. Only group manager can open a signature and find its originator. With the widespread applications of Internet, people pay more attention to their privacy since information technology can easily collect a person's private data without awareness of their owner. In recent years, group signature has attracted a lot of researchers and many schemes were proposed in the literature [17, 14, 15, 13, 1, 11, 12, 2, 10, 24, 7, 8]. A complete list of bibliography of group signatures can be found at [25]. It needs point out that the research of group signature has already evolved to more broad scopes in these days, such as anonymous identity and authentication. In such context, we treat anonymous signature as the proof that a member has been authenticated by a trusted third party [23, 9]. It is not regarded as a signature on behalf of a group any more.

Group signature is tightly coupled with its target applications compared with other cryptographic primitive such as encryption scheme. Therefore, the model of group signature scheme are some sort of informal in the literature and application oriented. In this paper, we follow the model in [10] which is a

relaxation to a strict definition proposed in [4]. This relaxation is mainly about group member revocation. To satisfy the requirements of the model in [4], it is impossible to revoke a group member except that all valid group members can somehow update their certificates. Our point of view about group signature is the preference of revocation mechanism. The reason essentially comes from the underlying assumption of group signature itself. We believe an underlying motivation of group signature is "distrust" assumption, i.e., people tend to doubt others would behave honestly. In group signature scheme, all group members are anonymous. If a corrupted member can not be identified, it is unlikely we would deploy such system. Otherwise we have to trust all group members would behave in expected manners. Thus comes the controversy of trust: group members do not trust others, and hope others would trust them. Therefore, a group signature without revocation mechanism has to assume all group members are honest, and this eventually conflicts with "distrust" assumption for a group signature. Therefore, we adopt the model in [10] to discuss our method to implement group signature.

Among group signature schemes in the literature, there are some constructions that share similar certificate structure, and are based on the same security assumption [13, 1, 12]. However, current schemes do not provide revocation mechanism. In this paper, we propose an effective method to integrate revocation mechanism into these constructions. We also give an example to demonstrate the method.

The paper is organized as follows. Section 2 reviews the definitions and security assumptions. In section 3 we introduce the proposed method. We apply this method to a well-known group signature scheme to implement efficient revocation mechanism in section 4. The paper concludes in section 5.

2 Definitions and Preliminaries

We adopt the model for group signature introduced in [10]. It is a relaxation of strict model in [4]. This model allows the revocation of a group member. As we discussed before, it is a more realistic model. We only outline core ideas of the model. Readers are encouraged to refer to [4, 10, 5] for formal treatment.

Definition 1 (The model). *A group signature scheme includes a group manager and group members. The group manager owns* group master keys *while each member holds its* group member key, or group member certificate. *The scheme consists of six protocols:*

- **KeyGen:** *the group manager uses KeyGen protocol to generate system parameters and its master key.*
- **Join:** *a party runs join protocol, together with the group manager, to obtain a certificate to represent its group membership.*
- **Sign:** *a group member anonymous sign a message following sign protocol.*
- **Verify:** *a verifier uses verify protocol to check whether a signature is originated from a member in the group.*

– **Open:** *the group manager uses open protocol to find the signer of a signature.*
– **Revoke:** *the group manager uses revoke protocol to exclude a group member.*

The security requirements for a group signature should have following properties:

– Full-traceability. *This property says that any valid signature can eventually be traced back to a legitimate group member. It should never happen that we cannot find the signer of a valid signature.* Full-traceability *has two implications: (1) a valid group certificate can only be created by the group manager, (2) a valid signature can only be generated by a legitimate group member if the secrets of the member is not exposed to any third party.*
– Anonymity. *This property says that if both the group manager's secret and a member's secret are not exposed, it is infeasible to find the signer of a signature, or link the signatures by a signer.*

The model in [4] defines *Full-Anonymity* which says even a member's secrets are exposed, it is still impossible to decide the signatures by this member. Obviously, under this strict model, we cannot revoke a member by exposing its secrets. Just as mentioned before, this property essentially precludes the possibility to revoke a group member explicitly.

Next, we review some definitions and widely accepted complexity assumptions that we will use in this paper.

Definition 2 (Special RSA modulus). *An RSA modulus $n = pq$ is called special if $p = 2p' + 1$ and $q = 2q' + 1$ where p' and q' also are prime numbers.*

Definition 3 (Quadratic Residue Group QR_n). *Let Z_n^* be the multiplicative group modulo n, which contains all positive integers less than n and relatively prime to n. An element $x \in Z_n^*$ is called a* quadratic residue *if there exists an $a \in Z_n^*$ such that $a^2 = x \,(\mathrm{mod} n)$. The set of all quadratic residues of Z_n^* forms a cyclic subgroup of Z_n^*, which we denote by QR_n. If n is the product of two distinct primes, then $|QR_n| = \frac{1}{4}|Z_n^*|$.*

Property 1 *If n is a special RSA modulus, with p, q, p', and q' as in Definition 2 above, then $|QR_n| = p'q'$ and $(p' - 1)(q' - 1)$ elements of QR_n are generators of QR_n.*

Property 2 *If g is a generator of QR_n, then $g^a \bmod n$ is a generator of QR_n if and only if $GCD(a, |QR_n|) = 1$.*

Property 3 *If $x \in_R Z_n^*$ is uniformly distributed over Z_n^*, then $x^2 \bmod n$ is uniformly distributed over QR_n.*

The security of our techniques relies on the following two assumptions, which are widely accepted in the cryptography literature (see, for example, [3, 19, 6]).

Assumption 1 (Strong RSA Assumption) *Let n be a special RSA modulus. The* Flexible RSA Problem *is the problem of taking a random element $u \in Z_n^*$ and finding a pair (v, e) such that $e > 1$ and $v^e = u \pmod{n}$. The* Strong RSA Assumption *says that no probabilistic polynomial time algorithm can solve the flexible RSA problem with non-negligible probability.*

Assumption 2 (Computational Diffie-Hellman Assumption for QR_n) *Let n be a special RSA modulus, and let g be a generator of QR_n. Then given random g^x and g^y, there is no probabilistic polynomial-time algorithm that computes $g^{xy} \pmod{n}$ with non-negligible probability.*

Assumption 3 (Decisional Diffie-Hellman Assumption for QR_n) *Let n be a special RSA modulus, and let g be a generator of QR_n. For two distributions (g, g^x, g^y, g^{xy}), (g, g^x, g^y, g^z), $x, y, x \in_R Z_n$, there is no probabilistic polynomial-time algorithm that distinguishes them with non-negligible probability.*

3 The Method to Implement Revocation Mechanism

In this section we introduce the method to implement revocation mechanism. We outline basic methodology without any real implementation. A specific construction depends on the requirements of target application, system parameters, and join protocol.

3.1 Group Member Certificate

Group signature schemes [13, 1, 12] are constructed over quadratic residue group QR_n where n is a special RSA modulus. The security of these schemes are based on the strong RSA assumption. In these schemes, a group certificate is in the form of

$$(A = g^{e^{-1}}, e),$$

where g is a generator of QR_n, and e^{-1} is the inverse of a prime number e modulo the order of QR_n. g could have some substructure such as $g = a^{x_i} a_0$ in [1].

3.2 Sign and Verify Protocols

To anonymously sign a message, a group member needs to hide its identity. It uses ElGamal encryption scheme [18] as

$$T_1 = Ay^w, T_2 = g^w,$$

where y is the group manager's ElGamal public key such that $y = g^x \mod n$. The group member also computes

$$T_3 = g^{we}.$$

A signer proves to a verifier that T_1, T_2, T_3 are constructed in such way that the hidden value A in T_1 is e-th root of g, and T_3 is the e-square of T_2. The building blocks for the proof are *statistical honest-verifier zero knowledge protocols of knowledge* related to discrete logarithm over QR_n [19, 20, 13]. They may include the protocols such as the knowledge of the discrete logarithm, the knowledge of equality of two discrete logarithms, the knowledge of the discrete logarithm that lies in certain interval, etc.

3.3 Group Member Revocation

We distinguish two group member revocation methods. The first one is to update all certificates for valid group members. In this way an excluded member indirectly loses its membership. We call it *Certificate Redistribution*. Another method is based on *Revocation List* which contains entries for the identification of revoked members. These two methods are complementary with each other.

In the context of group signature, due to the complex of join protocol, it is undesirable for valid group members to re-run join protocol to exclude a member implicitly. The problem is how to efficiently implement certificate redistribution without re-running join protocol. At the other side, frequently updating group certificates is not a satisfactory solution. We should be able to use revocation list to identify small number of revoked members. Ideally, a desirable solution would be first using revocation list to identify revoked members. When revocation list grows larger to a certain threshold, the group manager then updates all certificates for valid group members to exclude all revoked members, and revocation list is reset to empty. Our method can achieve such goal effectively.

We adopt certificate redistribution method which has been introduced in [2] without explicit addressing its security [1]. Recall that one component of a member certificate is $A = g^{e^{-1}}$. To update a valid certificate, the group manager picks a random integer r such that $GCD(r, |QR_n|) = 1$, computes

$$A' = A^r = g^{re^{-1}} = (g^r)^{e^{-1}} = g'^{e^{-1}}.$$

Due to *property 2*, g' will be another generator of QR_n. The group manager sends new certificates to valid group members in secure way.

It can be easily observed that this method does not need to re-run join protocol, and most computation can be completed by the group manager alone offline. This implies that the group manager can pre-compute all new certificates, and only distribute them to group members which are still legitimate. After certificate redistribution, each member still keeps its secrets. Consequently and significantly, redistribution could be done in nearly real-time.

This method has not been considered a good solution in [2]. However, we have a quite different opinion. We believe it is an effective method in many applications. The main negative comment about this method in [2] is that the group

[1] We independently devised the method in other place. Later, we noticed this method has already been introduced in [2]. However, our point of view to the method is quite positive.

manager needs to perform $O(n)$ cryptographic operations for every revoked member. However, this is not necessarily a bad thing in practice. In most situation, group manager may be server(s) with high computing capability. However, a group member could actually be a crypto-processor or smart card with limited resources, such as TPM (Trusted Platform module) in [23]. For the purpose of fast certificate redistribution, it is reasonable to let powerful servers undertake most computation task. In fact, any certificate redistribution method needs $O(n)$ operations. The real issues are about (1) the total computation overhead, and (2) how to distribute computation overhead among participants. Actually, some certificate redistribution methods just push the computation overhead to the group members [2, 10, 12], and some scheme has much higher total computation overhead. This may not be desirable if group members are resource limited.

Another advantage is the pre-computation of certificates. In the introduction we mentioned that group signature is not merely treated as an anonymous signature in these days. A lot of times we use it to implement anonymous access control/authentication in interactive manners [21, 1, 23]. In such application, when a corrupted member is being identified, the group manager needs immediately notify all verifiers (servers) that authentication should be based on some new system parameters. When a user tries to anonymously access a server, it will find access parameters have been updated. Then it needs to retrieve its new certificate from the group manager. Most of time, the group manager has already sent encrypted new certificates to valid members in secure way before a user notices system parameters have been changed. We have devised an encryption method which is similar to ElGamal encryption scheme. The group manager encrypts a new certificate A' as

$$(A^r,\ g^r A'),$$

where r is a large random integer in Z_n. Since $A = g^{e^{-1}}$, only group member itself can compute $(A^r)^e = g^r$, then it can further obtain A'. Most computation can be accomplished by the group manger in advance, without considering which group member is going to be excluded. And each valid group member only needs one decryption operation. It is even possible that the group manager can create multiple certificates for later use when a party joins the group. The pre-computation is a nice property. A seemingly inefficient method now becomes quite a good solution to fast certificate redistribution. We treat it is an effective method in certain context.

For an excluded group member, with existing certificate A that uses generator h, updating to a new certificate means computing $A' = g'^{e^{-1}} = g^{re^{-1}}$ based on $g^{e^{-1}}$ and $g' = g^r$ without knowing r or e^{-1}, which is equivalent to solving the computational Diffie-Hellman problem [2]. Therefore, we have the following theorem.

Theorem 1. *If there exists an algorithm that can compute an updated group member certificate without knowledge of the group manager's secret value, then*

[2] g' or certain substructure of g' will be published by the group manager according to a specific construction. Here we assume g' is being published.

there exists an algorithm that solves the computational Diffie-Hellman problem over QR_n.

Our method to implement revocation list is straightforward. The group manager puts a revoked member's e_i on list. To identify a revoked group member, a verifier checks

$$T_2^{e_i} =? T_3$$

for all e_i on the list. If the equation holds for one e_i, it shows the signature comes from a revoked member. This is a quite simple and efficient method (Of course, the list should be constrained to a reasonable size).

Remarks. It needs to point out that *revocation list* method implements *full revocation* defined in [10], or *unconditional linkability* defined in [2], i.e., all the signatures by a revoked member can be identified. Therefore a group signature scheme using this method only enjoys *anonymity*, not *full anonymity*. However, *certificate redistribution* method will not bring any issues related to anonymity. Therefore, in practice, we can adaptively choose either one to satisfy some criteria such as (1) system performance, or (2) privacy policy. For example, we may use revocation list to revoke members in one category, and certificate redistribution to preclude members in another category. The combination of these methods could provide us desirable flexibility.

3.4 Open Protocol

To open a signature, the group manager uses ElGamal decryption algorithm [18] to recover the identity of a group member

$$A = T_2^{-x} T_1.$$

4 A Real Example

In this section we give an example to show the effectiveness of the method in previous section. ACJT scheme is a well-known group signature construction introduced in 2000 [1]. It is a practical and provable secure construction for large group. However, it does not provide revocation mechanism. In the following we would like to adopt exact same notions as original paper. Thus, readers can easily compare the new scheme with original one, and see how our method integrates revocation mechanism into the original scheme.

We should notice that ACJT scheme achieves full anonymity without revocation mechanism, while new scheme provides revocation mechanism and achieves only anonymity. Again, we make it clear that this is an issue about how we are going to apply group signature to a specific application.

4.1 The System Parameters

- a special RSA modulus $n = pq$, $p = 2p' + 1$, $q = 2q' + 1$, p, p', q, q' are all prime.
- random elements $a, a_0, g \in QR_n$ of order $p'q'$, i.e., these numbers are the generators of QR_n.
- a random secret elements $x \in_R Z^*_{p'q'}$, and $y = g^x (\text{mod } n)$.
- security parameters used in protocols: $\epsilon > 1, k, l_p$.
- length parameters $\lambda_1, \lambda_2, \gamma_1, \gamma_2$. $\lambda_1 > \epsilon(\lambda_2+k)+2$, $\lambda_2 > 4l_p$, $\gamma_1 > \epsilon(\gamma_2+k)+2$, and $\gamma_2 > \lambda_1 + 2$.
- integer range $\Lambda =]2^{\lambda_1} - 2^{\lambda_2}, 2^{\lambda_1} + 2^{\lambda_2}[$ and $\Gamma =]2^{\gamma_1} - 2^{\gamma_2}, 2^{\gamma_1} + 2^{\gamma_2}[$.
- $H : \{0,1\}^* \rightarrow \{0,1\}^k$ is a strong collision-resistant hash function
- $m \in \{0,1\}^*$ is a message to be signed.
- the public parameters is (n, a, a_0, y, g).
- the secret parameters for the group manager is (p', q', x).

4.2 Join Protocol

We use the same join protocol as original scheme. A group member's certificate is in the form of $A_i = (a^{x_i} a_0)^{1/e_i} \mod n$ where $x_i \in \Lambda$ is the secret of a group member, and $e_i \in_R \Gamma$ is a random prime number. $a^{x_i} a_0$ can be seen as a generator of QR_n due to *property 1*. In new scheme, (A_i, e_i) MUST be kept secret by the group manager and a group member itself. In ACJT scheme, even though (A_i, e_i) is kept secret by the group manager and a group member, it would not affect the security of the scheme if it is publicly known due to its full anonymity property.

4.3 Sign Protocol

- Generate a random value $w \in_R \{0,1\}^{2l_p}$ and compute:

$$T_1 = A_i y^w \mod n, T_2 = g^w \mod n, T_3 = g^{we_i} \mod n.$$

- Randomly choose $r_1 \in_R \pm\{0,1\}^{\epsilon(\gamma_2+k)}, r_2 \in_R \pm\{0,1\}^{\epsilon(\lambda_2+k)}$, and $r_3 \in_R \pm\{0,1\}^{\epsilon(\lambda_1+2l_p+k+1)}$ and computes
  - $d_1 = T_1^{r_1}/(a^{r_2} y^{r_3}) \mod n, d_2 = T_2^{r_1}/g^{r_3} \mod n, d_3 = T_2^{r_1} \mod n$;
  - $c = H(g||y||a_0||a||T_1||T_2||T_3||d_1||d_2||d_3||m)$;
  - $s_1 = r_1 - c(e_i - 2^{\gamma_1}), s_2 = r_2 - c(x_i - 2^{\lambda_1}), s_3 = r_3 - ce_i w$ (all in Z_n).
- Output $(c, s_1, s_2, s_3, T_1, T_2, T_3)$.

Remarks. The main difference between new sign protocol and the original protocol is T_3, d_3. Our new method hide e_i as $T_2^{e_i}$. The original protocol in fact uses another ElGamal encryption to hide it as $g^{e_i} h^w$. r_4, d_4, s_4 in the original protocol are not used in new protocol. This roughly reduces thirty percent of computation overhead compared to the original protocol.

4.4 Verify Protocol

– Compute

$$c' = H(g||y||a_0||a||T_1||T_2||T_3||a_0^c T_1^{s_1 - c2^{\gamma_1}} / (a^{s_2 - c2^{\lambda_1}} y^{s_3}) || T_2^{s_1 - c2^{\gamma_1}} / g^{s_3} || T_2^{s_1 - c2^{\gamma_1}} T_3^c || m)$$

– Accept the signature if and only if $c = c'$ and $s_1 \in \pm\{0,1\}^{\epsilon(\gamma_2+k)+1}$, $s_2 \in \pm\{0,1\}^{\epsilon(\lambda_2+k)+1}$,$s_3 \in \pm\{0,1\}^{\epsilon(\lambda_1+2lp+k+1)+1}$.

Remarks. New verify protocol roughly reduces thirty percent of computation overhead compared to the original one.

4.5 Revocation Protocol

Based on idea introduced in previous section, the group manager picks a random large integer r such that $GCD(r, |QR_n|) = 1$, computes $a' = a^r, a'_0 = a_0^r$, and updates all certificates for valid group members to

$$A'_i = A_i^r = (a^r)^{e_i} a_0^r = a'^{e_i} a'_0 \mod n.$$

Group manager publish a', a'_0 and sends new certificates to valid group members in secure manner. This implements certificate redistribution.

To revoke a group member, the group manager adds e_i on revocation list and publish revocation list to all verifiers. A revoked group member can be identified by checking

$$T_2^{e_i} =? T_3 \mod n.$$

4.6 Security Properties of the New Protocol

Before discuss the security of the new scheme, we first introduce a lemma due to Shamir [22] that will be used shortly.

Lemma 1. *Let n be an integer. For given values $u, v \in Z_n^*$ and $x, y \in Z_n$ such that $GCD(x, y) = 1$ and $v^x = u^y \mod n$, there is an efficient way to compute the value z such that $z^x = u \mod n$.*

Proof. Since $GCD(x, y) = 1$, we can use the Extended GCD algorithm to find a and b such that $ay + bx = 1$, and let $z = v^a u^b$. Thus

$$z^x = v^{ax} u^{bx} = u^{ay+bx} = u \mod n.$$

□

Full-traceability is achieved by *zero knowledge* property of join protocol and coalition-resistance property of the group certificate which both have been proved in the original paper. We recall "coalition-resistance" property here.

Theorem 2 (Coalition-resistance). *Under the strong RSA assumption, a group certificate* $[A_i = (a^{x_i}a_0)^{1/e_i} \mod n, e_i]$ *with* $x \in \Lambda$ *and* $e_i \in \Gamma$ *can be generated only by the group manager provided that the number* K *of certificates the group manager issues is polynomially bounded.*

Next, we address the zero knowledge property of the group signature scheme. Since the new scheme has some difference with the original one, it is necessary to show it still keep this property. We recall the theorem in the original paper.

Theorem 3. *Under the strong RSA assumption, the interactive protocol underlying the group signature scheme is a statistical zero-knowledge (honest-verifier) proof of knowledge of a membership certificate and a corresponding membership secret key.*

Proof. Just as the original paper, we only address the proof of knowledge part. Considered that our construction has only minor difference with the original protocol, we only provide knowledge extracting method which is different from the original proof.

We should show that a knowledge extractor is able to recover the group certificate when it has found two accepting tuples under the same commitment and different challenges from a verifier. Let $(T_1, T_2, T_3, d_1, d_2, d_3, c, s_1, s_2, s_3)$ and $(T_1, T_2, T_3, d_1, d_2, d_3, c', s'_1, s'_2, s'_3)$ be such tuples.

Since $d_2 = T_2^{s_1 - c2^{\gamma_1}}/g^{s_3} = T_2^{s'_1 - c'2^{\gamma_1}}/g^{s'_3} \mod n$, we have

$$T_2^{(s'_1 - s_1) + (c - c')2^{\gamma_1}} = g^{s'_3 - s_3} \mod n.$$

If $GCD((s'_1 - s_1) + (c - c')2^{\gamma_1}, s'_3 - s_3) = k$, $k \neq 1$, then we have following equations for some v, v'.

$$(s'_1 - s_1) + (c - c')2^{\gamma_1} = kv, \ s'_3 - s_3 = kv',$$

$$(T_2^k)^v = g^{kv'} \mod n.$$

Since $GCD(kv', v) = 1$, due to lemma 1, we can find a solution (u, v) such that $u^v = g \mod n$. This is infeasible under the strong RSA assumption. Therefore, $(s'_1 - s_1) + (c - c')2^{\gamma_1}$ has to divide $s'_3 - s_3$, then we have

$$w = (s'_3 - s_3)/((s'_1 - s_1) + (c - c')2^{\gamma_1})$$

such that $T_2 = g^w \mod n$. Due the property of QR_n, T_2 is the generator of QR_n.

Since $d_3 = T_2^{s_1 - c2^{\gamma_1}} T_3^c = T_2^{s'_1 - c'2^{\gamma_1}} T_3^{c'} \mod n$, we have

$$T_2^{(s'_1 - s_1) + (c - c')2^{\gamma_1}} = T_3^{c - c'} \mod n.$$

Following the same method as above, under the strong RSA assumption, $c - c'$ has to divide $(s'_1 - s_1)$. We obtain

$$e_i = (s'_1 - s_1)/(c - c') + 2^{\gamma_1}$$

such that $T_3 = T_2^{e_i} \mod n$.

Based on the knowledge of w, e_i, we can further recover A_i, x_i the same way as the original proof. Therefore a knowledge extractor can fully recover group certificate. □

Anonymity property, NOT FULL anonymity in ACJT scheme, relies on the difficulty to decrypt T_1 which is encrypted by ElGamal encryption algorithm. Also it relies on the unlinkability of the two arbitrary signatures which has been proved in the original paper. Since we define a new T_3 in the new protocol, we need to show this modification still keep unlinkability property. Similar to the case in ACJT scheme, the problem of linking two tuples $(T_2, T_3), (T_2', T_3')$ reduces to decide the equality of the discrete logarithms of T_3, T_3' with base T_2, T_2' [3], respectively. This is assumed to be infeasible under the decisional Diffie-Hellman problem. Therefore, we have the following corollary.

Corollary 1. *Under the decisional Diffie-Hellman assumption for QR_n, there exists no probabilistic polynomial-time algorithm that can make the linkability decision for any two arbitrary tuples $(T_2, T_3), (T_2', T_3')$ with non-negligible probability.*

4.7 Related Works

After ACJT group signature scheme was introduced in 2000, several constructions have been proposed to integrate revocation mechanism to the scheme [2, 11, 24] . However, all these constructions are less efficient than the original one.

5 Conclusion

In this paper we introduced a generic method to integrate revocation into some group signature scheme. We demonstrated its effectiveness by applying this method to the well-known ACJT group signature scheme, and obtained a more efficient group signature scheme. This contrasts with other efforts such as [2, 11, 24] which result in less efficient constructions.

We should keep in mind the proposed method itself does not provide guarantee of the security. The security and performance rely on a specific construction. Also, a group signature scheme based on the method only achieves anonymity, not full anonymity in a more strict model. Even though, to implement efficient revocation, we do need anonymity property in practice.

References

1. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Advances in Cryptology — Crypto*, pages 255–270, 2000.

[3] The parameter settings in the scheme makes it infeasible to extract the discrete logarithm of T_2' with base T_2.

2. G. Ateniese, D. Song, and G.Tsudik. Quasi-efficient revocation in group signatures. In *Financial Cryptography'02*, pages 183–197, 2002.
3. N. Baric and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology — Eurocrypt*, pages 480–494, 1997.
4. M. Bellare, D. Micciancio, and B. Warinschi. Foundation of group signature: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Advances in Cryptology — EUROCRYPT'03, LNCS 2656*, pages 614–629, 2003.
5. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *Topics in Cryptology - CT-RSA 2005, LNCS 3376*, pages 136–153, 2005.
6. D. Boneh. The decision Diffie-Hellman problem. In *Proceedings of the Third Algorithmic Number Theory Symposium*, pages 48—63, 1998.
7. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Advances in Cryptology — Crypto'04, LNCS 3152*, pages 41–55, 2004.
8. D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *Proc. of the 11th ACM Conference on Computer and Communications Security (CCS 2004)*, pages 168–177, 2004.
9. E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *ACM Conference on Computer and Communications Security*, pages 132–145, 2004.
10. J. Camenisch and J. Groth. Group signatures: Better efficiency and new theoretical aspects. In *Security in Communication Networks (SCN 2004), LNCS 3352*, pages 120–133, 2005.
11. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Advances in Cryptology — Crypto'02, LNCS 2442*, pages 61–76, 2002.
12. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *SCN'02, LNCS 2576*, pages 268–289, 2002.
13. J. Camenisch and M. Michels. A group signature scheme based on an RSA-variants. Technical Report RS-98-27, BRICS, University of Aarhus, Nov. 1998.
14. J. Camenisch and M. Stadler. Efficient group signature schemems for large groups. In *Advances in Cryptology — Crypto'97, LNCS 1294*, pages 410–424, 1997.
15. J. Camenisch and M. Stadler. A group signature scheme with improved efficiency. In *Advances in Cryptology — ASIACRYPT'98, LNCS 1514*, pages 160–174, 1998.
16. D. Chaum and E. van Heyst. Group signature. In *Advances in Cryptology — Eurocrypt*, pages 390–407, 1992.
17. L. Chen and T. Pedersen. New group signatue schemes. In *Advances in Cryptology — EUROCRYPT'94, LNCS 950*, pages 171–181, 1995.
18. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology — Crypto*, pages 10–18, 1984.
19. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology — Crypto*, pages 16–30, 1997.
20. E. Fujisaki and T. Okamoto. A practical and provably secure scheme for publicly verifable secret sharing and its applications. In *Advances in Cryptology – EUROCRYPTO'98*, pages 32–46, 1998.
21. J. Kilian and E. Petrank. Identity escrow. In *Advances in Cryptology — Crypto*, pages 169–185, 1998.
22. A. Shamir. On the generation of cryptograpically strong psedorandom sequences. *ACM Transaction on computer systems*, 1, 1983.

23. TCG. http://www.trustedcomputinggroup.org.
24. G. Tsudik and S. Xu. Accumulating coposites and improved group signing. In *Advances in Cryptology — ASIACRYPT'03, LNCS 2894*, pages 269–286, 2003.
25. G. Wang. http://www.i2r.a-star.edu.sg/icsd/staff/guilin/bible/group-sign.htm.