An Effective Method to Implement Group Signature with Revocation

He Ge

Department of Computer Science and Engineering
University of North Texas
Denton, TX 76203

Abstract. This paper presents an effective method to integrate the revocation mechanism into some group signature schemes that are based on the strong RSA assumption. The mechanism enables the group manager to either update a group member's certificates, or revoke a group member. More specifically, a generic method has been proposed for the protocols of sign, verify, and revocation. We 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 schemes, strong RSA assumption, anonymity, full anonymity, revocation.

1 Introduction

A group signature scheme is a privacy-preserving cryptographic construction introduced by Chaum and Heyst in 1991 [16]. In such a scheme, a group member can sign a message on behalf of the group without revealing its identity. Only the group manager can open a signature and find its originator. In recent years, group signatures have attracted a lot of researchers, and many schemes have been proposed in the literature [14, 15, 13, 1, 11, 12, 2, 10, 21, 7, 8]. A complete list of bibliography of group signature schemes can be found at [22]. In practice, a group signature could be used to carry out anonymous authentication. That is, a signature verifier treats an anonymous signature as the proof that a party is a legitimate member in a group. Such applications have already been deployed in the Trusted Computing Platform [20, 9].

A group signature scheme is tightly coupled with its target applications compared with other cryptographic primitives (e.g. encryption scheme). The model for group signature schemes are context oriented. In this paper, we follow the model due to Camenisch and Joth [10], which is a relaxation to a strict definition proposed by Bellare *et al.* [4]. This relaxation is mainly about the group member revocation. To satisfy the requirements of the Bellare's model, it is impossible to revoke a group member except that all valid group members can somehow adjust the signing parameters or procedure, which may not always be feasible or efficient in practice. For the purpose of efficient revocation, many schemes (e.g. [2, 21, 8, 10]) have adopted the so-called "verifier-local revocation" technique, in which verifiers adjust their local verification parameters to recognize corrupted group members, and group members do not need to change the signing procedures at all. A group signature scheme based on this technique conforms to the relaxation mode by Camenisch and Joth.

Among the group signature schemes in the literature, there are some constructions that share similar group member certificate structure, and whose security is based on the same assumption [13, 1, 12]. However, these schemes do not provide any revocation mechanism. In this paper, we propose an effective method to integrate the 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. Section 3 introduces the proposed method. We apply this method to a well-known group signature scheme to implement an efficient revocation mechanism in section 4. The paper concludes in section 5.

2 Definitions and Preliminaries

We adopt the model for group signature by Camenish and Joth [10], a relaxation of the strict model by Bellare *et al.* [4]. Only the core ideas of the model will be introduced in this section. We refer the reader to [4, 10, 5] for a more in-depth discussion.

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 anonymously 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 member 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 are not exposed to any third party.*
- Anonymity. *This property says that if both the group manager's secrets and a member's secrets 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. 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 $(\mathrm{mod} n)$ is a generator of QR_n if and only if $GCD(a, |QR_n|) = 1$.*

The security of our techniques relies on the following assumptions, which are widely accepted in the cryptography literature (see, for example, [3, 18, 6]).

Assumption 1 (Strong RSA Assumption) *Let n be an 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 $v^e = u \,(\mathrm{mod}\, n)$. 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} 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), $x, y, z \in_R Z_n$, there is no probabilistic polynomial-time algorithm that distinguishes them with non-negligible probability.*

3 The Method to Implement Revocation Mechanism

This section introduces the method to carry out the revocation mechanism. We only outline the basic methodology without any real implementation. An exmple will be provided in the next section.

3.1 Group Member Certificate

The 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, e), where

$$A = g^{e^{-1}} \ (\mathrm{mod}\ n),$$

e is a prime number. g is a generator of QR_n, and e^{-1} is the inverse of e modulo the order of QR_n. g could have some substructure such as $g = a^{x_i} a_0 \ (\mathrm{mod}\ n)$ 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 [17] to compute

$$T_1 = Ay^w \pmod{n},\ T_2 = g^w \pmod{n},$$

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

$$T_3 = T_2^e \pmod{n}.$$

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 [18, 19, 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

To exclude a group member, the group manager broadcasts a revoked member's e_i to all verifiers. A verifier checks

$$T_2^{e_i} =?\ T_3 \pmod{n}$$

for all e_i on the revocation 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). This method is also called "verifier-local revocation" [8]. It needs to point out that the mechanism based on the revocation list 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*. This may seems a "weak point" for all group signature schemes with verifier-local revocation. However, the suitability of a feature really depends on a specific setting. There are no absolute arguments for the suitability of a feature in practice. For example, the anonymous authentication technique deployed in the Trusted Computing Platform, called Direct Anonymous Attestation, has adopted the verifier-local revocation method [20].

When the revocation list becomes large, the computation overhead may become unacceptable for verifiers. We introduce the certificate redistribution method proposed in [2]. We independently devised this method to implement key redistribution in the context of pervasive computing to reduce the computation overhead of resource-limited tiny electronic devices. To update a valid certificate, the group manager picks a random integer r such that $GCD(r, |QR_n|) = 1$, computing

$$A_i' = A_i^r = g^{re^{-1}} = (g^r)^{e^{-1}} = g'^{e^{-1}} \pmod{n}.$$

Due to *property 2*, g' is another generator of QR_n. The group manager sends new certificates to valid group members in secure way. The following operations then are based on the new system parameters and updated group member certificates. In this method, all the computation are accomplished by the group manager.

The authors in [2] have ignored the effectiveness of the method due to their arguments that the group manager needs to perform $O(n)$ cryptographic operations for every revoked member. In fact, it is easy to observe that 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. In many applications, the 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 the Trusted Computing Platform. In such settings, it is quite reasonable to let powerful servers undertake most computation task. In fact, some other certificate redistribution methods in [2, 10, 12] essentially push the computation to the group members, and have higher total computing overhead, which may not be desirable when the group members are resource-limited.

Furthermore, the group manager can pre-compute all certificates for group members offline. Only when the size of the revocation list passes a threshold, it publishes all the new certificates for valid group members. The pre-computation is a nice property which may be necessary to improve the system performance in practice.

For an excluded group member, with existing certificate A that uses generator g', 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 [1]. 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 there exists an algorithm that solves the computational Diffie-Hellman problem over QR_n.*

4 An Example

In this section we give an example to show the effectiveness of the method in the previous section. The 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 the revocation mechanism. In the following we would like to adopt the same notions as in the original paper. Thus, readers can easily compare the new scheme with the original one, seeing how our method carries out the revocation mechanism.

We should notice that the ACJT scheme achieves full anonymity without the revocation mechanism, while the new scheme provides the revocation mechanism and achieves only anonymity. Again, we make it clear that this is an issue about how we are going to apply a group signature scheme 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 $y = g^x (\mathrm{mod}\ n)$.
- security parameters used in protocols: $\epsilon > 1, k, l_p$.

[1] 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.

- 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 $\Gamma =]2^{\gamma_1} - 2^{\gamma_2}, 2^{\gamma_1} + 2^{\gamma_2}[$.
- $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^k$ 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 adopt the same Join protocol as in the original scheme. A group member's certificate is in the form of $A_i = (a^{x_i} a_0)^{1/e_i} \pmod{n}$ 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*.

One important difference between the new scheme and the ACJT scheme is that the Join procedure MUST be confidential. That is, in the new scheme, (A_i, e_i) MUST be kept secret by the group manager and a group member itself. In the ACJT scheme, it would not affect the security property of the scheme if (A_i, e_i) is publicly known. This is also the reason that the ACJT scheme achieves full anonymity.

4.3 Sign Protocol

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

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

- 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
  - $d_1 = T_1^{r_1}/(a^{r_2} y^{r_3}) \pmod{n}$, $d_2 = T_2^{r_1}/g^{r_3} \pmod{n}$, $d_3 = T_2^{r_1} \pmod{n}$;
  - $c = \mathcal{H}(g||y||a_0||a||T_1||T_2||T_3||d_1||d_2||d_3||m)$;
  - $s_1 = r_1 - c(e_i - 2^{\gamma_1})$, $s_2 = r_2 - c(x_i - 2^{\lambda_1})$, $s_3 = r_3 - ce_i w$ (all in Z_n).
- Output $(c, s_1, s_2, s_3, T_1, T_2, T_3)$.

Remark 1. The main difference between the new sign protocol and the original protocol is T_3, d_3. Our method hides e_i as $T_2^{e_i}$, while 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 needed in the new protocol, which roughly reduces thirty percent of computation overhead.

4.4 Verify Protocol

- Compute

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

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

4.5 Revocation Protocol

To revoke a corrupted group member, the group manager publishes e_i on the revocation list. A revoked group member can be identified by checking

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

In the meantime, the group manager picks a random large integer r such that $GCD(r, |QR_n|) = 1$, pre-computes $a' = a^r, a_0' = a_0^r$, and the certificates

$$A_i' = A_i^r = (a^{x_i r} a_0^r)^{1/e_i} = (a'^{x_i} a_0')^{1/e_i} \pmod n.$$

When the revocation list passes a threshold, the group manager publish a', a_0' and sends the new certificates to all valid group members in secure manners. This carries out certificate redistribution. At the same time, the revocation list is re-set to empty.

4.6 Security Properties of the New Protocol

Before discussing the security of the new scheme, we first introduce a lemma that will be used shortly.

Lemma 1. *Let n be an integer. Given values $u, v \in Z_n^*$ and $x, y \in Z$ such that $GCD(x, y) = r$, and $v^x \equiv u^y \pmod n$, there is an efficient way to compute a value z such that $z^k \equiv u \pmod n$, where $k = x/r$.*

Proof. Since $GCD(x, y) = r$, using the extended Euclidean GCD algorithm, we can obtain values α and β such that $\alpha x/r + \beta y/r = 1$. Then we have

$$u \equiv u^{\alpha x/r + \beta y/r} \equiv u^{\alpha x/r} u^{y\beta/r} \equiv u^{\alpha x/r} v^{\beta x/r}$$

$$\equiv (u^\alpha v^\beta)^{x/r} \pmod n.$$

Therefore, setting $k = x/r$ and $z = u^\alpha v^\beta$, we have $z^k \equiv u \pmod n$. □

Full-traceability is achieved by *zero knowledge* property of the 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 $[A_i = (a^{x_i} a_0)^{1/e_i} \pmod n, e_i]$ 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 new scheme. 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. 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 $d_2 \equiv T_2^{s_1 - c2^{\gamma_1}}/g^{s_3} \equiv T_2^{s'_1 - c'2^{\gamma_1}}/g^{s'_3} \pmod{n}$, we have

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

If $GCD((s'_1 - s_1) + (c - c')2^{\gamma_1}, s'_3 - s_3) = r$, $r \neq (s'_1 - s_1) + (c - c')2^{\gamma_1}$, By lemma 1, we can find a solution (u, v) such that $u^v = g \pmod{n}$. 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 $T_2 \equiv g^w \pmod{n}$. Due to the property of QR_n, T_2 is the generator of QR_n.

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

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

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 $T_3 \equiv T_2^{e_i} \pmod{n}$.

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

Unlinkability follows the same argument as the ACJT group signature for T_1, T_2. Since we define a different T_3 in our protocols, we need to show this modification still keep unlinkability property. Similar to the case in the 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 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.*

5 Conclusion

In this paper we have presented a generic method to integrate the revocation mechanism into some group signature schemes in the literature. We demonstrated its effectiveness by applying this method to the well-known the ACJT group signature scheme, and obtained a more efficient, and practical group signature scheme. This is in contrast to other efforts in [2, 11, 21], which result in less efficient constructions. The same method can also be applied to the group signature schemes in [13, 12].

It needs to point out that a group signature scheme based on our method only achieves anonymity, not full anonymity defined in the Bellare's strict model. However, in practice, anonymity is a more appropriate choice. Such level of privacy protection has been discussed in many research papers, for example, [2, 21, 8, 10, 9].

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.
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. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology — Crypto*, pages 10–18, 1984.
18. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology — Crypto*, pages 16–30, 1997.
19. 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.
20. TCG. http://www.trustedcomputinggroup.org.
21. G. Tsudik and S. Xu. Accumulating coposites and improved group signing. In *Advances in Cryptology — ASIACRYPT'03, LNCS 2894*, pages 269–286, 2003.
22. G. Wang. http://www.i2r.a-star.edu.sg/icsd/staff/guilin/bible/group-sign.htm.