Efficient and Provably Secure Trapdoor-Free Group Signature Schemes from Bilinear Pairings

Lan Nguyen and Rei Safavi-Naini

School of Information Technology and Computer Science,
University of Wollongong, Wollongong 2522, Australia
{ldn01, rei}@uow.edu.au

Abstract. We propose a group signature scheme with constant-size public key and signature length that does not require trapdoor. So system parameters can be shared by multiple groups belonging to different organizations. The scheme is provably secure in the formal model recently proposed by Bellare, Shi and Zhang (BSZ04), using random oracle model, Decisional Bilinear Diffie-Hellman and Strong Diffie-Hellman assumptions. We give a more efficient variant scheme and prove its security in a formal model which is a modification of BSZ04 model and has a weaker anonymity requirement. Both schemes are very efficient and the sizes of signatures are approximately one half and one third, respectively, of the sizes of the well-known ACJT00 scheme. We also use the schemes to construct a traceable signature scheme.

1 Introduction

Group signature schemes, introduced by Chaum and Van Heyst [14], allow a group member to sign a message on behalf of the group without revealing his identity and without allowing the message to be linkable to other signed messages that are verifiable with the same public key. Participants in a group signature scheme are a set of *group members* and a *group manager*. The role of the group manager is to register new users by issuing membership certificates that contain registration details, and in case of dispute about a signed message, revoking anonymity of the signed message by 'opening' the signature. In some schemes the functions of the group manager can be split between two managers: an *issuer* and an *opener*. This is a desirable property that allows distribution of trust. It is required that no collusion of the issuer and the opener can frame a group member. Group signatures are among the most important cryptographic primitives for providing privacy and have been used for applications such as anonymous credentials [2], identity escrow [21], voting and bidding [1], and electronic cash [23]. Kiayias et al. [18] also introduced the traceable signature primitive, which is basically the group signature system with added properties allowing a variety of levels for protecting user privacy.

In early group signature schemes [9, 14, 15] the size of the public key and the signature grew with the size of the group and so the schemes were impractical for large groups. Schemes with fixed size group public key and signature

P.J. Lee (Ed.): ASIACRYPT 2004, LNCS 3329, pp. 372–386, 2004.

length have been first proposed in [13] and later extended in [12, 1, 2]. In Crypto 2000, Ateniese et al. (ACJT00) [1] proposed an efficient group signature scheme with very short length and low computation cost. This scheme is also the only scheme that has been proved to satisfy the informal list of security requirements of group signature schemes. Ateniese and de Medeiros (AdM03) proposed an efficient group signature scheme [2] that is 'without trapdoor' in the sense that none of parties in the system including the group manager need to know the trapdoor. That is the system trapdoor is only used during the initialisation and to generate system parameters. The advantage of this property is that the same trapdoor information can be used to initiate different groups. The importance and usefulness of this property in real-world applications, for example when the group signature scheme is used as a building block of an anonymous credential system among a number of organizations that need to communicate and transfer information about users while protecting their privacy, have been outlined in [2]. A drawback of AdM03 scheme is that it has a single group manager who is responsible for registration of users and opening of signatures, and it is not possible to separate the two functionalities. In AdM03 scheme, the group manager stores the certificate (r, s) of each member. The signature of a group member contains elements χ and E_1 satisfying the equation $E_1 = \chi^r$, and so, to revoke a signature, the group manager (or any party with the knowledge of the certificates) can try all certificates to find the one satisfying the equation. This is an computationally expensive process. The security proof (corrected version) is for the informal list of security requirements, and is given in the generic model [3].

Security of a group signature scheme has been traditionally proved by showing that it satisfies a list of informally defined requirements. Bellare et al. [4] gave a formal security model (BSZ04) for (partially) dynamic groups with four security requirements (Correctness, Anonymity, Traceability and Non-frameability). The model uses various oracles including an Open oracle that takes a signed message and reveals the identity of the signer. The ACJT00 scheme although satisfies the conventional list of requirements but cannot be proved secure in the formal model mainly because of the inclusion of the Open oracle in the model. Kiayias et al. [19] proposed an extension (KY04 scheme) of ACJT00 scheme that is proved secure in their formal model. A new direction in constructing group signature schemes is to use bilinear pairings to shorten the lengths of the signature and key. Boneh et al. [7] proposed a short group signature scheme (BBS04) based on the Strong Diffie-Hellman assumption and a new assumption called the Decisional Linear assumption. The scheme is provably secure in a formal model where the Opening oracle is not available and the Non-frameability property is not required, in comparison with the BSZ04 model. They also showed how to construct an extension, which provides Non-frameability (exculpability). Based on the LRSW assumption [22], Camenisch and Lysyanskaya [11] proposed a group signature scheme (CL04) derived from a signature scheme which allows an efficient zero-knowledge proof of the knowledge of a signature on a committed message, and used it to construct an efficient anonymous credential system.

Our Contribution

In this paper, we first propose a new efficient group signature scheme with a number of attractive properties and prove its security in the BSZ04 model under the Decisional Bilinear Diffie-Hellman and Strong Diffie-Hellman assumptions, using random oracle model. We then give an efficient variant of this scheme and prove its security in the reduced version of BSZ04 model. The only difference between the original BSZ04 model and the reduced version is in modelling anonymity property, as in the reduced version, the adversary does not have access to the Open oracle. This is a plausible model for all cases that the opener is a highly trusted entity and cannot be accessed by the adversary. We also extend the variant scheme to a provably secure traceable signature scheme.

All proposed schemes have fixed lengths for group public key and signature, and so can be used for large size groups. Using elliptic curve cryptography in our schemes results in shorter lengths for signatures and keys. For example, for a comparable level of security as the ACJT00 scheme with 1024 bit composite modulus, our group signature schemes require elliptic curve groups of order 170 bit prime, resulting in the sizes of signatures in our two schemes to be one third and one half, respectively, of the size in ACJT00 scheme. For higher security levels this ratio will be smaller.

Our schemes can be converted into identity escrow systems or extended to support efficient membership revocation, as shown in [26]. The schemes are trapdoor-free. The only other trap-door free scheme is the AdM03 scheme, which uses a trapdoor in the initialisation of the system and assumes that the initialising party "safely forgets" the trapdoor. An advantage of our schemes over AdM03 scheme is that they allow separation of issuer and the opener, hence distribution of trust. Finally in our schemes, the interactive protocol underlying the signature scheme achieves honest verifier perfect zero-knowledge without any computational assumption whereas in the ACJT00 and KY04 schemes, the corresponding protocols achieve honest verifier statistical zero-knowledge under the Strong RSA assumption.

The paper is organized as follows. Section 2 gives related background and section 3 describes our group signature scheme and its security proofs. Section 4 gives a modification of BSZ04 formal model and a variant group signature scheme, and proves that the variant scheme and ACJT00 scheme are secure in the modified model. Section 5 describes our traceable signature scheme and section 6 provides efficiency comparison with ACJT00 scheme.

2 Preliminaries

2.1 Bilinear Pairings

Let $\mathbb{G}_1, \mathbb{G}_2$ be cyclic additive groups generated by P_1 and P_2, respectively, both with order p, a prime, and $\mathbb{G}_M$ be a cyclic multiplicative group with the same order. Suppose there is an isomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$ such that $\psi(P_2) = P_1$. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_M$ be a bilinear pairing with the following properties:

1. **Bilinearity:** $e(aP, bQ) = e(P, Q)^{ab}$ for all $P \in \mathbb{G}_1, Q \in \mathbb{G}_2, a, b \in \mathbb{Z}_p$
2. **Non-degeneracy:** $e(P_1, P_2) \neq 1$
3. **Computability:** There is an efficient algorithm to compute $e(P, Q)$ for all $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$

For simplicity, hereafter, we set $\mathbb{G}_1 = \mathbb{G}_2$ and $P_1 = P_2$ but our group signature schemes can be easily modified for the case when $\mathbb{G}_1 \neq \mathbb{G}_2$. For a group $\mathbb{G}$ of prime order, hereafter, we denote the set $\mathbb{G}^* = \mathbb{G} \backslash \{\mathcal{O}\}$ where $\mathcal{O}$ is the identity element of the group.

We define a Bilinear Pairing Instance Generator as a Probabilistic Polynomial Time (PPT) algorithm $\mathcal{G}$ that takes as input a security parameter 1^l and returns a uniformly random tuple $\mathbf{t} = (p, \mathbb{G}_1, \mathbb{G}_M, e, P)$ of bilinear pairing parameters, including a prime number p of size l, a cyclic additive group $\mathbb{G}_1$ of order p, a multiplicative group $\mathbb{G}_M$ of order p, a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_M$ and a generator P of $\mathbb{G}_1$.

2.2 Complexity Assumptions

For a function $f : \mathbb{N} \rightarrow \mathbb{R}^+$, if for every positive number α, there exists a positive integer l_0 such that for every integer $l > l_0$, it holds that $f(l) < l^{-\alpha}$, then f is said to be *negligible*. If there exists a positive number α_0 such that for every positive integer l, it holds that $f(l) < l^{\alpha_0}$, then f is said to be *polynomial-bound*.

The q-SDH assumption originates from a weaker assumption introduced by Mitsunari et. al. [24] to construct traitor tracing schemes [28] and later used by Zhang et al. [30] and Boneh et al. [5] to construct short signatures. It intuitively means that there is no PPT algorithm that can compute a pair $(c, \frac{1}{x+c}P)$, where $c \in \mathbb{Z}_p$, from a tuple $(P, xP, \ldots, x^qP)$, where $x \in_R \mathbb{Z}_p^*$.

q-Strong Diffie-Hellman (q-SDH) Assumption. *For every PPT algorithm $\mathcal{A}$, the following function $Adv_{\mathcal{A}}^{q\text{-}SDH}(l)$ is negligible.*

$$Adv_{\mathcal{A}}^{q\text{-}SDH}(l) = Pr[(\mathcal{A}(\mathbf{t}, P, xP, \ldots, x^qP) = (c, \frac{1}{x+c}P)) \wedge (c \in \mathbb{Z}_p)]$$

where $\mathbf{t} = (p, \mathbb{G}_1, \mathbb{G}_M, e, P) \leftarrow \mathcal{G}(1^l)$ *and* $x \leftarrow \mathbb{Z}_p^*$.

Intuitively, the DBDH assumption [6] states that there is no PPT algorithm that can distinguish between a tuple $(aP, bP, cP, e(P, P)^{abc})$ and a tuple (aP, bP, cP, Γ), where $\Gamma \in_R \mathbb{G}_M^*$ (i.e., chosen uniformly random from $\mathbb{G}_M^*$) and $a, b, c \in_R \mathbb{Z}_p^*$. It is defined as follows.

Decisional Bilinear Diffie-Hellman (DBDH) Assumption. *For every PPT algorithm $\mathcal{A}$, the following function $Adv_{\mathcal{A}}^{DBDH}(l)$ is negligible.*

$$Adv_{\mathcal{A}}^{DBDH}(l) = |Pr[\mathcal{A}(\mathbf{t}, aP, bP, cP, e(P, P)^{abc}) = 1] - Pr[\mathcal{A}(\mathbf{t}, aP, bP, cP, \Gamma) = 1]|$$

where $\mathbf{t} = (p, \mathbb{G}_1, \mathbb{G}_M, e, P) \leftarrow \mathcal{G}(1^l)$, $\Gamma \leftarrow \mathbb{G}_M^*$ *and* $a, b, c \leftarrow \mathbb{Z}_p^*$.

2.3 Bilinear Pairing Versions of El Gamal Public Key System

Based on the DBDH assumption, we can construct two bilinear pairing versions of El Gamal public key system. El Gamal^{BP1} provides Indistinguishability against adaptive Chosen Plaintext Attack (IND-CPA) and El Gamal^{BP2} provides Indistinguishability against adaptive Chosen Ciphertext Attack (IND-CCA) in the random oracle model. Due to space limitation, we only provide description of El Gamal^{BP2}. This is the bilinear pairing version of the scheme presented and proved by Fouque and Pointcheval [17]. Description of El Gamal^{BP1} can be found in the full version of this paper [25].

Key generation: Let $p, \mathbb{G}_1, \mathbb{G}_M, e$ be bilinear pairing parameters, as defined above, and G be a generator of $\mathbb{G}_1$. Suppose $x_a, x_b \in_R \mathbb{Z}_p^*$ and $\Theta_a = e(G,G)^{x_a}$ and $\Theta_b = e(G,G)^{x_b}$. The public key $pk = (G, \Theta_a, \Theta_b)$ and the secret key is $sk = (x_a, x_b)$. Choose a hash function $\mathcal{H}_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p$ (a random oracle).

Encryption: Plaintext $\Delta \in \mathbb{G}_M$ can be encrypted by choosing $t_a, t_b \in_R \mathbb{Z}_p^*$ and computing $(E_a, \Lambda_a) = (t_a G, \Delta\Theta_a^{t_a})$, $(E_b, \Lambda_b) = (t_b G, \Delta\Theta_b^{t_b})$ and a non-interactive zero-knowledge proof $\varsigma = (c, \rho_a, \rho_b)$ of equality of plaintexts between (E_a, Λ_a) and (E_b, Λ_b). The proof ς can be computed by choosing $w_a, w_b \in_R \mathbb{Z}_p$ and computing $c = \mathcal{H}_1(G||\Theta_a||\Theta_b||E_a||\Lambda_a||E_b||\Lambda_b||w_a G||w_b G||\Theta_a^{w_a}\Theta_b^{w_b})$, $\rho_a = w_a - t_a c$ and $\rho_b = w_b + t_b c$. The ciphertext is $(E_a, \Lambda_a, E_b, \Lambda_b, \varsigma)$.

Decryption: Given a ciphertext $(E_a, \Lambda_a, E_b, \Lambda_b, \varsigma)$, first check the validity of ς by verifying

$$c \stackrel{?}{=} \mathcal{H}_1(G||\Theta_a||\Theta_b||E_a||\Lambda_a||E_b||\Lambda_b||\rho_a G + cE_a||\rho_b G - cE_b||\Theta_a^{\rho_a}\Theta_b^{\rho_b}(\Lambda_a/\Lambda_b)^c)$$

then compute the plaintext $\Delta = \Lambda_a / e(E_a, G)^{x_a} = \Lambda_b / e(E_b, G)^{x_b}$.

Security: The security of El Gamal^{BP2} system is stated in Theorem 1.

Theorem 1. *El Gamal^{BP2} encryption scheme is IND-CCA if DBDH assumption holds, in the random oracle model.*

3 The Group Signature Scheme

3.1 Overview

Our group signature scheme is built upon two ordinary signature schemes. The first one is used in the Join, Iss protocol for the issuer to generate a signature (a_i, S_i) for each x_i, which is randomly generated by both a member and the issuer, but known only to the member. The second ordinary signature scheme is used in the GSig algorithm as the non-interactive version of a zero-knowledge protocol, that proves the signer's knowledge of (a_i, S_i) and x_i. The security of the two signature schemes underlies the security of the group signature scheme.

Our group signature scheme is constructed in cyclic groups with bilinear mappings. For simplicity, we present the scheme when the groups $\mathbb{G}_1$ and $\mathbb{G}_2$

are the same, however, it can be easily modified for the general case when $\mathbb{G}_1 \neq \mathbb{G}_2$. The users do not perform any pairing operation when signing, but pairing operation play an important role in the verification algorithm GVf. Intuitively, bilinear pairings allow a party, given $A, B, C, D \in \mathbb{G}_1$, to prove that $log_A B = log_C D$ without knowing $log_A B$ or $log_A C$. This is not possible in cyclic groups without bilinear pairings and where the DDH assumption holds.

3.2 Descriptions

We describe our group signature scheme according to the BSZ04 model, which is omitted in this paper due to space limitation. Our group signature scheme consists of two group managers (the issuer and the opener), and users with unique identities $i \in \mathbb{N}$ (the set of positive integers). Each user can join the group and become a group member. The scheme is specified as a tuple $\mathcal{GS}1 =$(GKg, UKg, Join, Iss, GSig, GVf, Open, Judge) of polynomial-time algorithms which are defined as follows. We assume that the group size and the number of queries asked by the adversary are polynomially-bounded by the security parameter l.

GKg: Suppose l is a security parameter and the Bilinear Pairing Instance Generator $\mathcal{G}$ generates a tuple of bilinear pairing parameters $\mathbf{t} = (p, \mathbb{G}_1, \mathbb{G}_M, e, P) \leftarrow \mathcal{G}(1^l)$, that is also the publicly shared parameters. Choose a hash function $\mathcal{H}_2 : \{0,1\}^* \rightarrow \mathbb{Z}_p$, which is assumed to be a random oracle in the security proofs. Choose $P_0, G, H \in_R \mathbb{G}_1$, $x, x'_a, x'_b \in_R \mathbb{Z}_p^*$ and compute $P_{pub} = xP$, $\Theta_a = e(G,G)^{x'_a}$ and $\Theta_b = e(G,G)^{x'_b}$. The group public key is $gpk = (P, P_0, P_{pub}, H, G, \Theta_a, \Theta_b)$, the issuing key is $ik = x$, and the opening key is $ok = (x'_a, x'_b)$.

UKg: This algorithm generates keys that provide authenticity for messages sent by the user in the (Join, Iss) protocol. This algorithm is the key generation algorithm K_S of any digital signature scheme (K_S, $Sign$, Ver) that is unforgeable against chosen message attacks (UNF-CMA). A user i runs the UKg algorithm that takes as input a security parameter 1^l and outputs a personal public and private signature key pair $(upk[i], usk[i])$. Public Key Infrastructure (PKI) can be used here. Although any UNF-CMA signature scheme can be used, but using schemes, whose security is based on DBDH or SDH assumptions, will reduce the underlying assumptions of our group signature scheme. One example of such scheme is in [5].

Join, Iss: In this protocol, a user i and the issuer first jointly generate a random value $x_i \in \mathbb{Z}_p^*$ whose value is only known by the user. The issuer then generates (a_i, S_i) for the user so that $e(a_i P + P_{pub}, S_i) = e(P, x_i P + P_0)$. The user uses $usk[i]$ to sign his messages in the protocol. Note that the formal model assumes the communication to be private and authenticated. We also assume that the communication is protected from replay attacks. The protocol is as follows.

1. user $i \longrightarrow$ issuer: $I = yP + rH$, where $y, r \in_R \mathbb{Z}_p^*$.
2. user $i \longleftarrow$ issuer: $u, v \in_R \mathbb{Z}_p^*$.
3. The user computes $x_i = uy + v$, $P_i = x_i P$.

4. user $i \longrightarrow$ issuer: P_i and a proof of knowledge of (x_i, r') such that $P_i = x_i P$ and $vP + uI - P_i = r'H$ (see [12] for this proof).
5. The issuer verifies the proof, then chooses $a_i \in_R \mathbb{Z}_p^*$ different from all corresponding elements previously issued, and computes $S_i = \frac{1}{a_i + x}(P_i + P_0)$.
6. user $i \longleftarrow$ issuer: a_i, S_i.
7. The user computes $\Delta_i = e(P, S_i)$, verifies if $e(a_i P + P_{pub}, S_i) = e(P, x_i P + P_0)$, and stores the *private signing key* $gsk[i] = (x_i, a_i, S_i, \Delta_i)$. Note that only the user knows x_i. The issuer also computes Δ_i and makes an entry in the table reg: $reg[i] = (i, \Delta_i, \langle \mathsf{Join}, \mathsf{Iss} \rangle$ transcript).

GSig: A group signature of a user i shows his knowledge of (a_i, S_i) and a secret x_i such that: $e(a_i P + P_{pub}, S_i) = e(P, x_i P + P_0)$. The signature does not reveal any information about his knowledge to anyone, except for the opener, who can compute Δ_i by decrypting an encryption of that value. The algorithm for a user i to sign a message $m \in \{0,1\}^*$ is as follows.

1. Encrypt Δ_i by El Gamal^{BP2} with public key (G, Θ_a, Θ_b) as $(E_a = tG, \Lambda_a = \Delta_i \Theta_a^t, E_b, \Lambda_b, \varsigma)$.
2. Perform the non-interactive version of a protocol, which we call the Signing protocol, as follows. Generate $r_1, ..., r_3, k_0, ..., k_5 \in_R \mathbb{Z}_p^*$ and compute
   (a) $U = r_1(a_i P + P_{pub})$; $V = r_2 S_i$; $W = r_1 r_2 (x_i P + P_0)$; $X = r_2 U + r_3 H$; $T_1 = k_1 P + k_2 P_{pub} + k_0 H$; $T_2 = k_3 P + k_2 P_0$; $T_3 = k_4 U + k_0 H$; $T_4 = k_5 G - k_4 E_a$; $\Pi = \Theta_a^{k_5} \Lambda_a^{-k_4}$.
   (b) $c = \mathcal{H}_2(gpk||E_a||\Lambda_a||E_b||\Lambda_b||\varsigma||U||V||W||X\ ||T_1||...||T_4||\Pi||m)$.
   (c) Compute in $\mathbb{Z}_p$: $s_0 = k_0 + cr_3$; $s_1 = k_1 + cr_1 r_2 a_i$; $s_2 = k_2 + cr_1 r_2$; $s_3 = k_3 + cr_1 r_2 x_i$; $s_4 = k_4 + cr_2$; $s_5 = k_5 + cr_2 t$.
3. Output the signature $(c, s_0, ..., s_5, U, V, W, X, E_a, \Lambda_a, E_b, \Lambda_b, \varsigma)$ for m.

GVf: The verification algorithm for $m, (c, s_0, ..., s_5, U, V, W, X, E_a, \Lambda_a, E_b, \Lambda_b, \varsigma)$ outputs accept if and only if verifying the proof ς outputs accept and the following two equations hold: $e(U, V) = e(P, W)$ and $c = \mathcal{H}_2(P||P_0||P_{pub}||H||G||\Theta||E_a||\Lambda_a||E_b||\Lambda_b||\varsigma||U||V||W||X||s_1 P + s_2 P_{pub} + s_0 H - cX||s_3 P + s_2 P_0 - cW||s_4 U + s_0 H - cX||s_5 G - s_4 E_a||\Theta_a^{s_5} \Lambda_a^{-s_4} e(P, cV)||m)$.

Open: To open m and its valid signature $(c, s_0, ..., s_5, U, V, W, X, E_a, \Lambda_a, E_b, \Lambda_b, \varsigma)$ to find the signer, the opener performs the following steps.

1. Use GVf algorithm to check the signature's validity. If the algorithm rejects, return $(0, \varepsilon)$, where ε denotes an empty string.
2. Compute $\Delta_i = \Lambda_a e(E_a, G)^{-x'_a}$ and find the corresponding entry i in the table reg. If no entry is found, return $(0, \varepsilon)$.
3. Return $reg[i]$ and a non-interactive zero-knowledge proof ϱ of knowledge of x'_a so that $\Theta_a = e(G, G)^{x'_a}$ and $\Lambda_a / \Delta_i = e(E_a, G)^{x'_a}$ (see [12] for this proof).

Judge: On an output by the Open algorithm for a message m and its signature ω, the Judge algorithm is performed as follows:

1. If Open algorithm outputs $(0, \varepsilon)$, run GVf algorithm on m, ω. If GVf rejects, return accept; otherwise, return reject.
2. If Open algorithm outputs $(reg[i], \varrho)$, return reject if one of the following happens: (i) on m, ω, GVf algorithm rejects; (ii) verification of the proof ϱ rejects; (iii) the $\langle$Join, Iss$\rangle$ transcript is invalid with regard to $upk[i]$; (iv) $\Delta_i \neq e(P, S_i)$ where S_i is extracted from the $\langle$Join, Iss$\rangle$ transcript. Otherwise, return accept.

Remarks:

- Our scheme is trapdoor-free. This improves efficiency and manageability, and various groups can share the same initial set-up $p, \mathbb{G}_1, \mathbb{G}_M, e, P, P_0, G, H$.
- Our Signing protocol achieves honest verifier perfect zero-knowledge and does not rely on any complexity assumption. This indicates a higher level of unconditional security: from a signature, an adversary with unlimited power (but without access to the *reg* table) can compute only a part of the signer's registration information (S_i), whereas, in the ACJT00 and KY04 schemes, the adversary can find all parts of the signer's private signing key.

3.3 Security Proofs

Theorem 2. *The group signature scheme $\mathcal{GS}1$ provides Correctness.*

Theorem 3. *The group signature scheme $\mathcal{GS}1$ provides Anonymity in the random oracle model if the Decisional Bilinear Diffie-Hellman assumption holds.*

Theorem 4. *The group signature scheme $\mathcal{GS}1$ provides Traceability in the random oracle model if the q-Strong Diffie-Hellman assumption holds, where q is the upper bound of the group size.*

Theorem 5. *The group signature scheme $\mathcal{GS}1$ provides Non-frameability in the random oracle model if the Discrete Logarithm assumption holds over the group $\mathbb{G}_1$ and the digital signature scheme $(K_S, Sign, Ver)$ is UNF-CMA.*

Proofs of these theorems can be found in the full version [25]. We provide here the proofs of two important properties that underlie these theorems, i. e. the Zero-knowledge property of the Signing protocol in GSig algorithm and the Coalition-Resistance of $\mathcal{GS}1$ and $\mathcal{GS}2$. In our definition, Coalition-Resistance intuitively means that a colluding group of signers, with the knowledge of the opening key and access to some oracles, should not be able to generate a new valid user private signing key. For a group signature scheme $\mathcal{GS}$, a PPT adversary $\mathcal{A}$, a PPT predicate $\mathcal{U}$ that can determine the validity of a user private signing key, and any security parameter $l \in \mathbb{N}$, the formula of the experiment for Coalition-Resistance is as follows.

Experiment $Exp^{\mathsf{coal.re}}_{\mathcal{GS},\mathcal{A},\mathcal{U}}(l)$
$(gpk, ik, ok) \leftarrow \mathsf{GKg}(1^l)$; $\mathsf{CU} \leftarrow \emptyset$; $\mathsf{HU} \leftarrow \emptyset$
$gsk' \leftarrow \mathcal{A}(gpk, ok : \mathsf{CrptU}(\cdot,\cdot), \mathsf{SndToI}(\cdot,\cdot), \mathsf{AddU}(\cdot), \mathsf{RReg}(\cdot), \mathsf{USK}(\cdot))$
If $gsk' \in \{gsk[i] |\ i \in \mathsf{CU} \cup \mathsf{HU}\}$ then return 0 else return $\mathcal{U}(gpk, gsk')$

HU is a set of honest users; CU - a set of corrupted users; GSet - a set of message-signature pairs ; AddU$(\cdot)$ - add user oracle; CrptU$(\cdot,\cdot)$ - corrupt user oracle; SndToI$(\cdot,\cdot)$ - send to issuer oracle; USK$(\cdot)$ - user secret keys oracle; RReg$(\cdot)$ - read registration table oracle. The group signature scheme $\mathcal{GS}$ provides Coalition-Resistance if the following function $Adv^{\mathsf{coal.re}}_{\mathcal{GS},A,\mathcal{U}}(l)$ is negligible.

$$Adv^{\mathsf{coal.re}}_{\mathcal{GS},A,\mathcal{U}}(l) = \mathsf{Pr}[Exp^{\mathsf{coal.re}}_{\mathcal{GS},A,\mathcal{U}}(l) = 1]$$

Lemma 1. *The interactive Signing protocol underlying the GSig algorithm is a (honest-verifier) perfect zero-knowledge proof of knowledge of (a_i, S_i), x_i and t such that $e(a_iP + P_{pub}, S_i) = e(P, x_iP + P_0)$, $E_a = tG$ and $\Lambda_a = e(P, S_i)\Theta_a^t$.*

Proof. The proof for completeness is straightforward. The proofs of Soundness and Zero-knowledge property are as follows.

Soundness: If the protocol accepts with non-negligible probability, we show that the prover must have the knowledge of (a_i, S_i), x_i and t satisfying the relations stated in the theorem. Suppose the protocol accepts for the same commitment $(U, V, W, X, T_1, ..., T_4, \Pi)$, two different pairs of challenges and responses $(c, s_0, ...s_5)$ and $(c', s'_0, ..., s'_5)$. Let $f_i = \frac{s_i - s'_i}{c - c'}, i = 0, ..., 5$, then: $X = f_1P + f_2P_{pub} + f_0H$; $W = f_3P + f_2P_0$; $X = f_4U + f_0H$; $E_a = f_5f_4^{-1}G$; $e(P, V) = \Theta_a^{-f_5}\Lambda_a^{f_4}$; so $U = f_1f_4^{-1}P + f_2f_4^{-1}P_{pub}$.

Let $a_i = f_1f_2^{-1}$, $S_i = f_4^{-1}V$, $x_i = f_3f_2^{-1}$, $t = f_5f_4^{-1}$, then $E_a = tG$, $\Lambda_a = e(P, S_i)\Theta_a^t$ and $e(a_iP + P_{pub}, S_i) = e(P, x_iP + P_0)$, as $e(U, V) = e(P, W)$. So the prover have the knowledge of (a_i, S_i), x_i and t satisfying the relations.

Zero-knowledge: The simulator chooses $c, s_0, ...s_5 \in_R \mathbb{Z}_p$, $b \in_R \mathbb{Z}_p^*$, $X, V \in_R \mathbb{G}_1$ and compute $U = bP$, $W = bV$, $T_1 = s_1P + s_2P_{pub} + s_0H - cX$, $T_2 = s_3P + s_2P_0 - cW$, $T_3 = s_4U + s_0H - cX$, $T_4 = s_5G - s_4E_a$ and $\Pi = \Theta_a^{s_5}\Lambda_a^{-s_4}e(P, cV)$. We can see that the distribution of the simulation is the same as the distribution of the real transcript.

Lemma 2. *If the q-SDH assumption holds, then the group signature schemes $\mathcal{GS}1$ and $\mathcal{GS}2$, whose group sizes are bounded by q, provide Coalition-Resistance, where the predicate $\mathcal{U}$ is defined as:*

$$\mathcal{U}(\langle P, P_0, P_{pub}, ...\rangle, \langle x_i, a_i, S_i, \Delta_i\rangle) = 1 \Leftrightarrow e(a_iP + P_{pub}, S_i) = e(P, x_iP + P_0).$$

Proof. We prove the lemma for both $\mathcal{GS}1$ and $\mathcal{GS}2$. Suppose there is a PPT adversary $\mathcal{A}$ that can break the Coalition-Resistance property of $\mathcal{GS}1$ or $\mathcal{GS}2$ with respect to the predicate $\mathcal{U}$ defined above. Let the set of private signing keys generated during $\mathcal{A}$'s attack be $\{(x_i, a_i, S_i, \Delta_i)\}_{i=1}^q$ and let his output be a new private signing key $(x^*, a^*, S^*, \Delta^*)$ with non-negligible probability (that means $(a^*, S^*) \notin \{(a_i, S_i)\}_{i=1}^q$). We show a construction of a PPT adversary $\mathcal{B}$ that can break the q-SDH assumption. Suppose a tuple $challenge = (Q, zQ, \ldots, z^qQ)$ is given, where $z \in_R \mathbb{Z}_p^*$; we show that $\mathcal{B}$ can compute $(c, 1/(z+c)Q)$, where $c \in \mathbb{Z}_p$ with non-negligible probability. We consider two cases.

Case 1: This is a trivial case, where $\mathcal{A}$ outputs $S^* \in \{S_1, ..., S_q\}$ with non-negligible probability. In this case, $\mathcal{B}$ chooses $x, x'_a, x'_b \in_R \mathbb{Z}_p^*$ and $G, H \in_R \mathbb{G}_1$, gives $\mathcal{A}$ the group signature public key $(P = Q, P_0 = zQ, P_{pub} = xP, H, G, \Theta_a = e(G,G)^{x'_a}, \Theta_b = e(G,G)^{x'_b})$ and the opening key (x'_a, x'_b) (no x'_b, Θ'_b in case of $\mathcal{GS}2$), and simulates a set of possible users. Then $\mathcal{B}$ can simulate all oracles that $\mathcal{A}$ needs to access. Suppose a set of private signing keys $\{(x_i, a_i, S_i, \Delta_i)\}_{i=1}^q$ is generated and $\mathcal{A}$ outputs a new $(x^*, a^*, S^*, \Delta^*)$ with non-negligible probability such that $S^* \in \{S_1, ..., S_q\}$. Suppose $S^* = S_j$, where $j \in \{1, ..., q\}$, then $\frac{1}{a^*+x}(x^*P + P_0) = \frac{1}{a_j+x}(x_jP + P_0)$, so $(a_j - a^*)P_0 = (a^*x_j - a_jx^* + x_jx - x^*x)P$. Therefore, z is computable by $\mathcal{B}$ from this, and so is $(c, 1/(z+c)Q)$, for any $c \in \mathbb{Z}_p$.

Case 2: This is when the first case does not hold. That means $\mathcal{A}$ outputs $S^* \notin \{S_1, ..., S_q\}$ with non-negligible probability. Then $\mathcal{B}$ plays the following game:

1. Generate $\alpha, a_i, x_i \in_R \mathbb{Z}_p^*$, $i = 1, ..., q$, where a_is are different from one another, then choose $m \in_R \{1, ..., q\}$.
2. Let $x = z - a_m$ ($\mathcal{B}$ does not know x), then the following P, P_{pub}, P_0 are computable by $\mathcal{B}$ from the tuple *challenge*.

$$P = \prod_{i=1, i\neq m}^{q} (z + a_i - a_m)Q$$

$$P_{pub} = xP = (z - a_m) \prod_{i=1, i\neq m}^{q} (z + a_i - a_m)Q$$

$$P_0 = \alpha \prod_{i=1}^{q} (z + a_i - a_m)Q - x_m \prod_{i=1, i\neq m}^{q} (z + a_i - a_m)Q$$

3. Generate $x'_a, x'_b \in_R \mathbb{Z}_p^*$ and $G, H \in_R \mathbb{G}_1$ and give $\mathcal{A}$ the group signature public key $(P, P_0, P_{pub}, H, G, \Theta_a = e(G,G)^{x'_a}, \Theta_b = e(G,G)^{x'_b})$ and the opening key (x'_a, x'_b) (no x'_b, Θ'_b in case of $\mathcal{GS}2$) and simulates a set of possible users.
4. With the capabilities above, $\mathcal{B}$ can simulate oracles $\mathsf{CrptU}(\cdot,\cdot)$, $\mathsf{RReg}(\cdot)$ and $\mathsf{USK}(\cdot)$) that $\mathcal{A}$ needs to access. For $\mathsf{AddU}(\cdot)$ or $\mathsf{SndToI}(\cdot,\cdot)$, $\mathcal{B}$ simulates the addition of an honest or corrupted user i as follows. As playing both sides of the Join, Iss protocol or being able to extract information from $\mathcal{A}$, $\mathcal{B}$ simulates the protocol as specified so that the prepared a_i, x_i above are computed in the protocol to be the corresponding parts of the user i's private signing key. $\mathcal{B}$ can compute S_i as follows:
   - If $i = m$, then $S_m = \frac{1}{a_m+x}(x_mP + P_0) = \alpha \prod_{i=1, i\neq m}^{q} (z + a_i - a_m)Q$. This is computable from the tuple *challenge*.
   - If $i \neq m$, then $S_i = \frac{1}{a_i+x}(x_iP + P_0) = (x_i - x_m) \prod_{j=1, j\neq m, i}^{q} (z + a_j - a_m)Q + \alpha \prod_{j=1, j\neq i}^{q} (z + a_j - a_m)Q$. This is computable from the tuple *challenge*.
5. Get the output $(x^*, a^*, S^*, \Delta^*)$ from $\mathcal{A}$, where $S^* = \frac{1}{a^*+x}(x^*P + P_0) = \frac{1}{z+a^*-a_m}(\alpha z + x^* - x_m) \prod_{i=1, i\neq m}^{q} (z + a_i - a_m)Q$

We can see that the case $\alpha z+x^*-x_m = \alpha(z+a^*-a_m)$ happens with negligible probability, as it results in $S^* = S_m$. So the case $\alpha z + x^* - x_m \neq \alpha(z + a^* - a_m)$ happens with non-negligible probability ϵ_1. Suppose in this case, the probability that $a^* \in \{a_1, ..., a_q\}$ is ϵ_2. Then the probability that $a^* \notin \{a_1, ..., a_q\}\backslash\{a_m\}$ is $\epsilon_1 - \frac{q-1}{q}\epsilon_2$ (as $m \in_R \{1, ..., q\}$), which is also non-negligible if q is polynomially bound by the security parameter l. If $\alpha z + x^* - x_m \neq \alpha(z + a^* - a_m)$ and $a^* \notin \{a_1, ..., a_q\}\backslash\{a_m\}$, then $\frac{1}{z+a^*-a_m}Q$ is computable from the tuple *challenge* and S^* and so $\mathcal{B}$ can compute $(c, \frac{1}{z+c}Q)$, where $c = a^* - a_m$.

4 Variations

4.1 Weak Anonymity Requirement

We introduce this security requirement to account for a class of group signature schemes, including ACJT00 scheme, which can not be proved to achieve Anonymity requirement. Weak Anonymity requirement is defined exactly the same as Anonymity requirement, except that the adversary does not have access to the $\mathsf{Open}(\cdot,\cdot)$ oracle. In practice, when the opener is assumed to be uncorrupted as in Anonymity requirement, it could be hard for the adversary to have access to the Open oracle. As Open oracle is not used in the conventional list of requirements, the same argument as in [4] shows that Weak anonymity, Traceability and Non-frameability are sufficient to imply the conventional list of requirements.

4.2 A Variant Group Signature Scheme, $GS2$

The scheme $\mathcal{GS}2$ is the same as $\mathcal{GS}1$, except that in the signature, Δ_i is encrypted by El Gamal^{BP1} encryption scheme instead of El Gamal^{BP2}. So in GKg, x'_b and Θ_b are not generated and in GSig, Δ_i is encrypted by El Gamal^{BP1} public key (G, Θ_a) as $(E_a = tG, \Lambda_a = \Delta_i\Theta_a^t)$. So there is no E_b, Λ_b or ς in the signature and in the executions of GSig, GVf, Open and Judge algorithms. Security of $\mathcal{GS}2$ is stated in Theorem 6, whose proof is shown in the full version [25].

Theorem 6. *$\mathcal{GS}2$ provides Correctness. $\mathcal{GS}2$ provides Weak Anonymity if the Decisional Bilinear Diffie-Hellman assumption holds. $\mathcal{GS}2$ provides Traceability in the random oracle model if the q-Strong Diffie-Hellman assumption holds, where q is the upper bound of the group size. $\mathcal{GS}2$ provides Non-frameability in the random oracle model if the Discrete Logarithm assumption holds over the group $\mathbb{G}_1$ and the digital signature scheme $(K_S, Sign, Ver)$ is UNF-CMA.*

4.3 Do ACJT00 and $GS2$ Schemes Provide Anonymity?

We first state the security of the ACJT00 scheme in Theorem 7. The ACJT00 scheme refers to the scheme proposed in [1], plus some simple extensions to accommodate the Judge algorithm (defining the UKg algorithm as in our scheme, using $usk[i]$ to sign messages in the Join, Iss protocol, and verifying signatures in the Open and Judge algorithms). The methodology of the proof for Theorem

7 is very similar to the proof of Theorem 6, and the exact details of each step can be extracted from the proofs in [19].

Theorem 7. *The ACJT00 scheme provides Correctness; Weak Anonymity if the DDH-Compo-KF assumption holds; Traceability in the random oracle model if the Strong RSA assumption holds; Non-frameability in the random oracle model if the Discrete Logarithm assumption holds over the quadratic residues group of a product of two known large primes, and the digital signature scheme for* UKg *is UNF-CMA. (See [19] for assumptions used in this theorem).*

It is an open question if the ACJT00 and $\mathcal{GS}2$ schemes provide Anonymity, in line with the open problem whether a combination of an El Gamal encryption (IND-CPA) and a Schnorr proof of knowledge of the plaintext can provide IND-CCA. This combination has been proved to provide IND-CCA in the random oracle model, but the proof has required either another very strong assumption [29] or is in generic model [27]. In ACJT00 and $\mathcal{GS}2$ signatures, the identity-bound information is encrypted by variations of El Gamal encryption and the other part of the signatures proves knowledge of the information. The Open oracle plays a similar role as the Decryption oracle in the model of IND-CCA.

4.4 Variants Based on the DDH Assumption

We can build variants of $\mathcal{GS}1$ and $\mathcal{GS}2$, whose security is based on the DDH assumption over the group $\mathbb{G}_M$ instead of the DBDH (DDHV) assumption. Specifically, Δ_i will be encrypted by the normal El Gamal encryption scheme or the twin-paradigm extension of El Gamal encryption scheme (proposed in [17]). The Open algorithm in these variant schemes requires one less pairing operation than in $\mathcal{GS}1$ and $\mathcal{GS}2$.

We can actually provide a group signature with 4 options, where the users, the issuer and the opener use the same keys for all options. The first two options are $\mathcal{GS}1$ and $\mathcal{GS}2$, offering smaller signature size and more efficient signing and verification. The last two options are the variant schemes based on the normal DDH assumption, with more efficient opening.

5 A Traceable Signature Scheme

We extend $\mathcal{GS}2$ to be a traceable signature scheme $\mathcal{TS}$ =(Setup, Join, Sign, Verify, Open, Reveal, Trace, Claim, $\mathsf{Claim\text{-}Verify}$) with similar advantages over the only other traceable signature scheme [18].

Setup: This is the same as GKg for $\mathcal{GS}2$, but the group public key also includes a $Q \in_R \mathbb{Z}_p^*$. The group public key is $gpk = (P, P_0, P_{pub}, Q, H, G, \Theta_a)$, the issuing key is $ik = x$, and the opening key is $ok = x'_a$. Choose a hash function $\mathcal{H}_3 : \{0,1\}^* \rightarrow \mathbb{Z}_p$ (a random oracle).

Join: This protocol is the same as the Join, Iss protocol in Section 3.2, except for the following. The GM also chooses $\bar{x_i} \in_R \mathbb{Z}_p^*$, computes $S_i = \frac{1}{a_i+x}(P_i+\bar{x_i}Q+P_0)$

at step 5 and sends the user $a_i, S_i, \bar{x}_i$ at step 6. In the last step, the user computes $\Delta_i = e(P, S_i)$, verifies if $e(a_i P + P_{pub}, S_i) = e(P, x_i P + \bar{x}_i Q + P_0)$, and stores the *private signing key* $gsk[i] = (x_i, \bar{x}_i, a_i, S_i, \Delta_i)$. The GM also computes Δ_i and stores it with the protocol's transcript.

Sign: The algorithm for an user i to sign a message $m \in \{0,1\}^*$ is as follows.

1. Compute $E_a = tG$, $\Lambda_a = \Delta_i \Theta_a^t$, $\Upsilon_1 = \Theta_a^{\bar{x}_i r}$, $\Upsilon_2 = \Theta_a^r$, $\Upsilon_3 = \Theta_a^{x_i r'}$ and $\Upsilon_4 = \Theta_a^{r'}$, where $t, r, r' \in_R \mathbb{Z}_p^*$.
2. Generate $r_1, ..., r_3, k_0, ..., k_6 \in_R \mathbb{Z}_p^*$ and compute
   (a) $U = r_1(a_i P + P_{pub})$; $V = r_2 S_i$; $W = r_1 r_2 (x_i P + \bar{x}_i Q + P_0)$; $X = r_2 U + r_3 H$; $T_1 = k_1 P + k_2 P_{pub} + k_0 H$; $T_2 = k_3 P + k_6 Q + k_2 P_0$; $T_3 = k_4 U + k_0 H$; $T_4 = k_5 G - k_4 E_a$; $\Pi = \Theta_a^{k_5} \Lambda_a^{-k_4}$; $\Psi_1 = \Upsilon_1^{-k_2} \Upsilon_2^{k_6}$; $\Psi_2 = \Upsilon_3^{-k_2} \Upsilon_4^{k_3}$.
   (b) $c = \mathcal{H}_3(P||P_0||P_{pub}||H||G||\Theta||E_a||\Lambda_a||E_b||\Lambda_b||\varsigma||U||V||W||X||T_1||...||T_4 ||\Pi||\Psi_1||\Psi_2||m)$.
   (c) Compute in $\mathbb{Z}_p$: $s_0 = k_0 + cr_3$; $s_1 = k_1 + cr_1 r_2 a_i$; $s_2 = k_2 + cr_1 r_2$; $s_3 = k_3 + cr_1 r_2 x_i$; $s_4 = k_4 + cr_2$; $s_5 = k_5 + cr_2 t$; $s_6 = k_6 + cr_1 r_2 \bar{x}_i$
3. Output the signature $(c, s_0, ..., s_6, U, V, W, X, E_a, \Lambda_a, \Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4)$ for m.

Verify: The verification algorithm for $m, (c, s_0, ..., s_6, U, V, W, X, E_a, \Lambda_a, \Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4)$ outputs accept if and only if the following two equations hold: (i) $e(U, V) = e(P, W)$ and (ii) $c = \mathcal{H}_3(P||P_0||P_{pub}||H||G||\Theta||E_a||\Lambda_a||E_b||\Lambda_b||\varsigma||U||V||W||X ||s_1 P + s_2 P_{pub} + s_0 H - cX||s_3 P + s_6 Q + s_2 P_0 - cW||s_4 U + s_0 H - cX||s_5 G - s_4 E_a ||\Theta_a^{s_5} \Lambda_a^{-s_4} e(P, cV)||\Upsilon_1^{-s_2} \Upsilon_2^{s_6}||\Upsilon_3^{-s_2} \Upsilon_4^{s_3}||m)$

Open: To open m and its valid signature $(c, s_0, ..., s_5, U, V, W, X, E_a, \Lambda_a, \Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4)$ to find the signer, the GM computes $\Delta_i = \Lambda_a e(E_a, G)^{-x'_a}$ and finds the corresponding entry i in the table of stored Join transcripts. The GM returns i and a non-interactive zero-knowledge proof ϱ of knowledge of x'_a so that $\Theta_a = e(G, G)^{x'_a}$ and $\Lambda_a / \Delta_i = e(E_a, G)^{x'_a}$ (see [12] for this proof).

Reveal and Trace: Given the Join transcript of user i, the GM recovers the tracing trapdoor $trace_i = \bar{x}_i$. Given $trace_i$ and a message-signature pair, a designated party recovers Υ_1 and Υ_2 and checks if $\Upsilon_1 = \Upsilon_2^{\bar{x}_i}$. If the equation holds, the tracer concludes that user i has produced the signature.

Claim and Claim-Verify: Given a message-signature pair, a user i can claim that he is the signer by recovering Υ_3 and Υ_4 and producing a non-interactive proof of knowledge of the discrete-log of Υ_3 base Υ_4. Any party can run Claim-Verify by verifying the signature and the proof.

Security. The security of $\mathcal{TS}$ is stated in Theorem 8. The proof of this theorem uses techniques similar to those in [18] and arguments similar to the proofs for our group signature schemes.

Theorem 8. *In the random oracle model, $\mathcal{TS}$ provides (i) security against misidentification attacks based on the q-SDH and the DDH assumptions, where q is the upper bound of the group size; (ii) security against anonymity attacks*

based on the DBDH and DDH assumptions; (iii) security against framing attacks based on the DL assumption.

6 Efficiency

The sizes of signatures and keys in our schemes are much shorter than those used in the Strong-RSA-based schemes at a similar level of security. This difference grows when higher level of security is required. In this section, we compare sizes in our new group signature schemes with those in ACJT00 scheme. We assume that our scheme is implemented using an elliptic curve or hyperelliptic curve over a finite field. p is a 170-bit prime, $\mathbb{G}_1$ is a subgroup of an elliptic curve group or a Jacobian of a hyperelliptic curve over a finite field of order p. $\mathbb{G}_M$ is a subgroup of a finite field of size approximately 2^{1024}. A possible choice for these parameters can be found in [8], where $\mathbb{G}_1$ is derived from the curve $E/GF(3^\iota)$ defined by $y^2 = x^3 - x + 1$. We assume that system parameters in ACJT00 scheme are $\epsilon = 1.1$, $l_p = 512$, $k = 160$, $\lambda_1 = 838$, $\lambda_2 = 600$, $\gamma_1 = 1102$ and $\gamma_2 = 840$. We summarize the result in Table 1.

Table 1. Comparison of sizes (in Bytes)

	Signature	gpk	gsk	ik	ok	Security
ACJT00	1087	768	370	128	128	Weak Anonymity
$\mathcal{GS}1$	597	363	192	22	44	Anonymity
$\mathcal{GS}2$	384	235	192	22	22	Weak Anonymity

Acknowledgements. Authors thank anonymous referees of Asiacrypt 2004 for constructive comments and Fangguo Zhang for helpful discussions.

References

1. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. CRYPTO 2000, Springer-Verlag, LNCS 1880, pp. 255-270.
2. G. Ateniese, and B. de Medeiros. Efficient Group Signatures without Trapdoors. ASIACRYPT 2003, Springer-Verlag, LNCS 2894, pp. 246-268.
3. G. Ateniese, and B. de Medeiros. Security of a Nyberg-Rueppel Signature Variant. Cryptology ePrint Archive, Report 2004/093, http://eprint.iacr.org/.
4. M. Bellare, H. Shi, and C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. Cryptology ePrint Archive: Report 2004/077.
5. D. Boneh, and X. Boyen. Short Signatures Without Random Oracles. EUROCRYPT 2004, Springer-Verlag, LNCS 3027, pp. 56-73.
6. D. Boneh, and X. Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. EUROCRYPT 2004, Springer-Verlag, LNCS 3027, pp. 223-238.

7. D. Boneh, X. Boyen, and H. Shacham. Short Group Signatures. CRYPT0 2004, Springer-Verlag, LNCS, to appear.
8. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. ASIACRYPT 2001, Springer-Verlag, LNCS 2248, pp.514-532.
9. J. Camenisch. Efficient and generalized group signatures. EUROCRYPT 1997, Springer-Verlag, LNCS 1233, pp. 465-479.
10. J. Camenisch, and A. Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. CRYPTO 2002, Springer-Verlag, LNCS 2442, pp. 61-76.
11. J. Camenisch, and A. Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. CRYPTO 2004, Springer-Verlag, LNCS, to appear.
12. J. Camenisch, and M. Michels. A group signature scheme with improved efficiency. ASIACRYPT 1998, Springer-Verlag, LNCS 1514.
13. J. Camenisch, and M. Stadler. Efficient group signature schemes for large groups. CRYPTO 1997, Springer-Verlag, LNCS 1296.
14. D. Chaum, and E. van Heyst. Group signatures. CRYPTO 1991, LNCS 547, Springer-Verlag.
15. L. Chen, and T. P. Pedersen. New group signature schemes. EUROCRYPT 1994, Springer-Verlag, LNCS 950, pp. 171-181.
16. A. Fiat, and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. CRYPTO 1986, Springer-Verlag, LNCS 263, pp. 186-194.
17. P. Fouque and D. Pointcheval, Threshold Cryptosystems Secure against Chosen-Ciphertext Attacks, ASIACRYPT 2001, Springer-Verlag, LNCS 2248, pp. 351-368.
18. A. Kiayias, Y. Tsiounis and M. Yung. Traceable Signatures. EUROCRYPT 2004, Springer-Verlag, LNCS 3027, pp. 571-589.
19. A. Kiayias, and Moti Yung. Group Signatures: Provable Security, Efficient Constructions and Anonymity from Trapdoor-Holders. Cryptology ePrint Archive: Report 2004/076.
20. J. Killian, and E. Petrank. Identity escrow. CRYPTO 1998, Springer-Verlag, LNCS 1642, pp. 169-185.
21. S. Kim, S. Park, and D. Won. Convertible group signatures. ASIACRYPT 1996, Springer-Verlag, LNCS 1163, pp. 311-321.
22. A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. SAC 1999, Springer-Verlag, LNCS 1758.
23. M. Michels. Comments on some group signature schemes. TR-96-3-D, Department of Computer Science, University of Technology, Chemnitz-Zwickau, Nov. 1996.
24. S. Mitsunari, R. Sakai, and M. Kasahara. A new traitor tracing. IEICE Trans. Vol. E85-A, No.2, pp. 481-484, 2002.
25. L. Nguyen, and R. Safavi-Naini. Efficient and Provably Secure Trapdoor-free Group Signature Schemes from Bilinear Pairings. Full version.
26. L. Nguyen. Accumulators from Bilinear Pairings and Applications. CT-RSA 2005, Springer-Verlag, LNCS, to appear.
27. P. Schnorr and M. Jakobsson. Security of signed El Gamal encryption. ASIACRYPT 2000, Springer-Verlag, LNCS 1976, pp. 73-89.
28. V. To, R. Safavi-Naini, and F. Zhang. New traitor tracing schemes using bilinear map. DRM Workshop 2003.
29. Y. Tsiounis and M. Yung. On the security of El Gamal based encryption. PKC 1998, Springer-Verlag, LNCS 1431, pp. 117-134.
30. F. Zhang, R. Safavi-Naini and W. Susilo. An Efficient Signature Scheme from Bilinear Pairings and Its Applications. PKC 2004, Springer-Verlag, LNCS 2947, pp. 277-290.