A Practical and Provably Secure Coalition-Resistant Group Signature Scheme

[Published in M. Bellare, Ed., *Advances in Cryptology – CRYPT0 2000*, vol. 1880 of *Lecture Notes in Computer Science*, pp. 255–270, Springer-Verlag, 2000.]

Giuseppe Ateniese[1], Jan Camenisch[2], Marc Joye[3], and Gene Tsudik[4]

[1] Department of Computer Science, The Johns Hopkins University
3400 North Charles Street, Baltimore, MD 21218, USA
ateniese@cs.jhu.edu
[2] IBM Research, Zurich Research Laboratory
Säumertrasse 4, CH-8803 Rüschlikon, Switzerland
jca@zurich.ibm.com
[3] Gemplus Card International, Card Security Group
Parc d'Activités de Gémenos, B.P. 100, F-13881 Gémenos, France
marc.joye@gemplus.com
[4] Department of Information and Computer Science,
University of California, Irvine, Irvine, CA 92697-3425, USA
gts@ics.uci.edu

Abstract. A group signature scheme allows a group member to sign messages anonymously on behalf of the group. However, in the case of a dispute, the identity of a signature's originator can be revealed (only) by a designated entity. The interactive counterparts of group signatures are identity escrow schemes or group identification scheme with revocable anonymity. This work introduces a new provably secure group signature and a companion identity escrow scheme that are significantly more efficient than the state of the art. In its interactive, identity escrow form, our scheme is proven secure and coalition-resistant under the strong RSA and the decisional Diffie-Hellman assumptions. The security of the non-interactive variant, i.e., the group signature scheme, relies additionally on the Fiat-Shamir heuristic (also known as the random oracle model).

Keywords. Group signature schemes, revocable anonymity, coalition-resistance, strong RSA assumption, identity escrow, provable security.

1 Introduction

Group signature schemes are a relatively recent cryptographic concept introduced by Chaum and van Heyst [CvH91] in 1991. In contrast to ordinary signatures they provide anonymity to the signer, i.e., a verifier can only tell that a member of some group signed. However, in exceptional cases such as a legal dispute, any group signature can be "opened" by a designated group manager

to reveal unambiguously the identity of the signature's originator. At the same time, no one — including the group manager — can misattribute a valid group signature.

The salient features of group signatures make them attractive for many specialized applications, such as voting and bidding. They can, for example, be used in invitations to submit tenders [CP95]. All companies submitting a tender form a group and each company signs its tender anonymously using the group signature. Once the preferred tender is selected, the winner can be traced while the other bidders remain anonymous. More generally, group signatures can be used to conceal organizational structures, e.g., when a company or a government agency issues a signed statement. Group signatures can also be integrated with an electronic cash system whereby several banks can securely distribute anonymous and untraceable e-cash. This offers concealing of the cash-issuing banks' identities [LR98].

A concept dual to group signature schemes is identity escrow [KP98]. It can be regarded as a group-member identification scheme with revocable anonymity. A group signature scheme can be turned into an identity escrow scheme by signing a random message and then proving the knowledge of a group signature on the chosen message. An identity escrow scheme can be turned into a group signature scheme using the Fiat-Shamir heuristic [FS87]. In fact, most group signature schemes are obtained in that way from 3-move honest-verifier proof of knowledge protocols.

This paper presents a new group signature / identity escrow scheme that is provably secure. In particular, the escrow identity scheme is provably coalition-resistant under the strong RSA assumption. Other security properties hold under the decisional Diffie-Hellman or the discrete logarithm assumption. Our group signature scheme is obtained from the identity escrow scheme using the Fiat-Shamir heuristic, hence it is secure in the random oracle model.

Our new (group signature) scheme improves on the state-of-the-art exemplified by the scheme of Camenisch and Michels [CM98a] which is the only known scheme whose coalition-resistance is provable under a standard cryptographic assumption. In particular, our scheme's registration protocol (`JOIN`) for new members is an order of magnitude more efficient. Moreover, our registration protocol is statistically zero-knowledge with respect to the group member's secrets. In contrast, in [CM98a] the group member is required to send the group manager the product of her secret, a prime of special form, and a random prime; such products are in principle susceptible to an attack due to Coppersmith [Cop96]. Moreover, our scheme is provably coalition-resistance against an adaptive adversary, whereas for the scheme by Camenisch and Michels [CM98a] this holds only for a static adversary.

The rest of this paper is organized as follows. The next section presents the formal model of a secure group signature scheme. Section 3 overviews cryptographic assumptions underlying the security of our scheme and introduces some basic building blocks. Subsequently, Section 4 presents the new group signature scheme. The new scheme is briefly contrasted with prior work in Section 5. The security properties are considered in Section 6. Finally, the paper concludes in Section 7.

2 The Model

Group-signature schemes are defined as follows. (For an in-depth discussion on this subject, we refer the reader to [Cam98].)

Definition 1. *A* group-signature scheme *is a digital signature scheme comprised of the following five procedures:*

`SETUP`: *On input a security parameter ℓ, this probabilistic algorithm outputs the initial group public key $\mathcal{Y}$ (including all system parameters) and the secret key $\mathcal{S}$ for the group manager.*

`JOIN`: *A protocol between the group manager and a user that results in the user becoming a new group member. The user's output is a membership certificate and a membership secret.*

`SIGN`: *A probabilistic algorithm that on input a group public key, a membership certificate, a membership secret, and a message m outputs group signature of m.*

`VERIFY`: *An algorithm for establishing the validity of an alleged group signature of a message with respect to a group public key.*

`OPEN`: *An algorithm that, given a message, a valid group signature on it, a group public key and a group manager's secret key, determines the identity of the signer.*

A secure group signature scheme must satisfy the following properties:

Correctness: Signatures produced by a group member using `SIGN` must be accepted by `VERIFY`.

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

Anonymity: Given a valid signature of some message, identifying the actual signer is computationally hard for everyone but the group manager.

Unlinkability: Deciding whether two different valid signatures were computed by the same group member is computationally hard.

Exculpability: Neither a group member nor the group manager can sign on behalf of other group members.[1] A closely related property is that of *nonframing* [CP95]; it captures the notion of a group member not being made responsible for a signature she did not produce.

Traceability: The group manager is always able to open a valid signature and identify the actual signer.

Note that the last property is also violated if a subset of group members, pooling together their secrets, can generate a valid group signature that cannot be opened by the group manager. Because this was ignored in many papers we state it explicitly as an additional property.

[1] Note that the above does not preclude the group manager from creating fraudulent signers (i.e., nonexistent group members) and then producing group signatures.

Coalition-resistance: A colluding subset of group members (even if comprised of the entire group) cannot generate a valid signature that the group manager cannot link to one of the colluding group members.

We observe that many group signature schemes (e.g., [CM98a,CM99b,CS97]) can be viewed as making use of two different ordinary signature schemes: one to generate membership certificates as part of `JOIN` and another to actually generate group signatures as part of `SIGN` (cf. [CM99b]). Consequently, the properties of any secure group signature scheme must include the Unforgeability property (as defined in [GMR88]) for each of the two ordinary signature schemes. It is easy to see that each of: Traceability and Exculpability map into the Unforgeability property for the two respective signature schemes. Furthermore, together they ensure that a group signature scheme is unforgeable, i.e., that only group members are able to sign messages on behalf of the group.

The model of identity escrow schemes [KP98] is basically the same as the one for group signature schemes; the only difference being that the `SIGN` algorithm is replaced by an interactive protocol between a group member and a verifier.

3 Preliminaries

This section reviews some cryptographic assumptions and introduces the building blocks necessary in the subsequent design of our group signature scheme. (It can be skipped with no significant loss of continuity.)

3.1 Number-Theoretic Assumptions

The *Strong-RSA Assumption* (SRSA) was independently introduced by Barić and Pfitzmann [BF97] and by Fujisaki and Okamoto [FO97]. It strengthens the widely accepted RSA Assumption that finding e^{th}-roots modulo n — where e is the public, and thus fixed, exponent — is hard to the assumption that finding an e^{th}-root modulo n for any $e > 1$ is hard. We give hereafter a more formal definition.

Definition 2 (Strong-RSA Problem). *Let $n = pq$ be an RSA-like modulus and let G be a cyclic subgroup of $\mathbb{Z}_n^*$ of order $\#G$, $\lceil \log_2(\#G) \rceil = \ell_G$. Given n and $z \in G$, the* Strong-RSA Problem *consists in finding $u \in G$ and $e \in \mathbb{Z}_{>1}$ satisfying $z \equiv u^e \pmod{n}$.*

Assumption 1 (Strong-RSA Assumption). *There exists a probabilistic polynomial-time algorithm $\mathcal{K}$ which on input a security parameter ℓ_G outputs a pair (n, z) such that, for all probabilistic polynomial-time algorithms $\mathcal{P}$, the probability that $\mathcal{P}$ can solve the Strong-RSA Problem is negligible.*

The *Diffie-Hellman Assumption* [DH76] appears in two "flavors": (i) the Computational Diffie-Hellman Assumption (CDH) and (ii) the Decisional Diffie-Hellman Assumption (DDH). For a thorough discussion on the subject we refer the reader to [Bon98].

Definition 3 (Decisional Diffie-Hellman Problem). *Let $G = \langle g \rangle$ be a cyclic group generated by g of order $u = \#G$ with $\lceil \log_2(u) \rceil = \ell_G$. Given g, g^x, g^y, and $g^z \in G$, the* Decisional Diffie-Hellman Problem *consists in deciding whether the elements g^{xy} and g^z are equal.*

This problem gives rise to the *Decisional Diffie-Hellman Assumption*, which was first explicitly mentioned in [Bra93] by Brands although it was already implicitly assumed in earlier cryptographic schemes.

Assumption 2 (Decisional Diffie-Hellman Assumption). *There is no probabilistic polynomial-time algorithm that distinguishes with non-negligible probability between the distributions D and R, where $D = (g, g^x, g^y, g^z)$ with $x, y, z \in_R \mathbb{Z}_u$ and $R = (g, g^x, g^y, g^{xy})$ with $x, y \in_R \mathbb{Z}_u$.*

The Decisional Diffie-Hellman Problem is easier than the *(Computational) Diffie-Hellman Problem* which involves finding g^{uv} from g^u and g^v; the Decisional Diffie-Hellman Assumption is, thus, a *stronger* assumption. Both are stronger assumptions than the assumption that computing discrete logarithms is hard.

If n is a *safe* RSA modulus (i.e., $n = pq$ with $p = 2p' + 1$, $q = 2q' + 1$, and p, q, p', q' are all prime), it is a good habit to restrict operation to the subgroup of quadratic residues modulo n, i.e., the cyclic subgroup $\mathrm{QR}(n)$ generated by an element of order $p'q'$. This is because the order $p'q'$ of $\mathrm{QR}(n)$ has no small factors.

The next corollary shows that it is easy to find a generator g of $\mathrm{QR}(n)$: it suffices to choose an element $a \in \mathbb{Z}_n^*$ satisfying $\gcd(a \pm 1, n) = 1$ and then to take $g = a^2 \bmod n$. We then have $\mathrm{QR}(n) = \langle g \rangle$. (By convention, $\gcd(0, n) := n$.)

Proposition 1. *Let $n = pq$, where $p \neq q$, $p = 2p' + 1$, $q = 2q' + 1$, and p, q, p', q' are all prime. The order of the elements in $\mathbb{Z}_n^*$ are one of the set $\{1, 2, p', q', 2p', 2q', p'q', 2p'q'\}$. Moreover, if the order of $a \in \mathbb{Z}_n^*$ is equal to $p'q'$ or $2p'q' \iff \gcd(a \pm 1, n) = 1$.* □

Corollary 1. *Let n be as in Proposition 1. Then, for any $a \in \mathbb{Z}_n^*$ s.t. $\gcd(a \pm 1, n) = 1$, $\langle a^2 \rangle \subset \mathbb{Z}_n^*$ is a cyclic subgroup of order $p'q'$.* □

Remark 1. Notice that $4\ (= 2^2)$ always generates $\mathrm{QR}(n)$ whatever the value of a safe RSA modulus n. Notice also that the Jacobi symbol $(g|n) = +1$ does *not* necessarily imply that g is a quadratic residue modulo n but merely that $(g|p) = (g|q) = \pm 1$, where $(g|p)$ (resp. $(g|q)$) denotes the Legendre symbol[2] of g modulo p (resp. q). For example, $(2|55) = (2|5)\,(2|11) = (-1)\,(-1) = +1$; however, there is no integer x such that $x^2 \equiv 2 \pmod{55}$.

Deciding whether some y is in $\mathrm{QR}(n)$ is generally believed infeasible if the factorization of n is unknown.

[2] By definition, the Legendre symbol $(g|p) = +1$ if g is a quadratic residue modulo p, and -1 otherwise.

3.2 Signatures of Knowledge

So-called zero-knowledge proofs of knowledge allow a prover to demonstrate the knowledge of a secret w.r.t. some public information such that no other information is revealed in the process. The protocols we use in the following are all 3-move protocols and can be proven zero-knowledge in an honest-verifier model. Such protocols can be performed non-interactively with the help of an ideal hash function $\mathcal{H}$ (*à la* Fiat-Shamir [FS87]). Following [CS97], we refer to the resulting constructs as *signatures of knowledge*. One example is the Schnorr signature scheme [Sch91] where a signature can be viewed as a proof of knowledge of the discrete logarithm of the signer's public key made non-interactive.

In the following, we consider three building blocks: signature of knowledge of (i) a discrete logarithm; (ii) equality of two discrete logarithms; and (iii) a discrete logarithm lying in a given interval. All of these are constructed over a cyclic group $G = \langle g \rangle$ the order of which $\#G$ is unknown; however its bit-length ℓ_G (i.e., the integer ℓ_G s.t. $2^{\ell_G - 1} \leq \#G < 2^{\ell_G}$) is publicly known. Fujisaki and Okamota [FO97] show that, under the SRSA, the standard proofs of knowledge protocols that work for a group of known order are also proofs of knowledge in this setting. We define the *discrete logarithm* of $y \in G$ w.r.t. base g as any integer $x \in \mathbb{Z}$ such that $y = g^x$ in G. We denote $x = \log_g y$. We assume a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^k$ which maps a binary string of arbitrary length to a k-bit hash value. We also assume a security parameter $\epsilon > 1$.

Showing the knowledge of the discrete logarithm of $y = g^x$ can be done easily in this setting as stated by the following definition (cf. [Sch91]).

Definition 4. *Let $y, g \in G$. A pair $(c, s) \in \{0,1\}^k \times \pm\{0,1\}^{\epsilon(\ell_G + k)+1}$ verifying $c = \mathcal{H}(y \| g \| g^s\, y^c \| m)$ is a signature of knowledge of the discrete logarithm of $y = g^x$ w.r.t. base g, on a message $m \in \{0,1\}^*$.*

The party in possession of the secret $x = \log_g y$ is able to compute the signature by choosing a random $t \in \pm\{0,1\}^{\epsilon(\ell_G + k)}$ and then computing c and s as:

$$c = \mathcal{H}(y \| g \| g^t \| m) \quad \text{and} \quad s = t - cx \quad (\text{in } \mathbb{Z}) \ .$$

A slight modification of the previous definition enables to show the knowledge and equality of two discrete logarithms of, say y_1 and y_2, with bases g and h, i.e., knowledge of an integer x satisfying $y_1 = g^x$ and $y_2 = h^x$.

Definition 5. *Let $y_1, y_2, g, h \in G$. A pair $(c, s) \in \{0,1\}^k \times \pm\{0,1\}^{\epsilon(\ell_G + k)+1}$ verifying $c = \mathcal{H}(y_1 \| y_2 \| g \| h \| g^s\, {y_1}^c \| h^s\, {y_2}^c \| m)$ is a signature of knowledge of the discrete logarithm of both $y_1 = g^x$ w.r.t. base g and $y_2 = h^x$ w.r.t. base h, on a message $m \in \{0,1\}^*$.*

The party in possession of the secret x is able to compute the signature, provided that $x = \log_g y_1 = \log_h y_2$, by choosing a random $t \in \pm\{0,1\}^{\epsilon(\ell_G + k)}$ and then computing c and s as:

$$c = \mathcal{H}(y_1 \| y_2 \| g \| h \| g^t \| h^t \| m) \quad \text{and} \quad s = t - cx \quad (\text{in } \mathbb{Z}) \ .$$

In Definition 4, a party shows the knowledge of the discrete logarithm of y w.r.t. base g. The order of g being unknown, this means that this party knows an integer x satisfying $y = g^x$. This latter condition may be completed in the sense that the party knows a discrete logarithm x lying in a given interval. It is a slight modification of a protocol appearing in [FO98].

Definition 6. *Let $y, g \in G$. A pair $(c, s) \in \{0,1\}^k \times \pm\{0,1\}^{\epsilon(\ell+k)+1}$ verifying $c = \mathcal{H}(y \,\|g\, \|g^{s-cX}\, y^c \,\|m)$ is a signature of knowledge of the discrete logarithm $\log_g y$ that lies in $]X - 2^{\epsilon(\ell+k)}, X + 2^{\epsilon(\ell+k)}[$, on a message $m \in \{0,1\}^*$.*

From the knowledge of $x = \log_g y \in]X - 2^\ell, X + 2^\ell[$, this signature is obtained by choosing a random $t \in \pm\{0,1\}^{\epsilon(\ell+k)}$ and computing c and s as:

$$c = \mathcal{H}(y\|g\|g^t\|m),\ s = t - c(x - X) \quad (\text{in } \mathbb{Z}) \ .$$

Remark 2. Note that, although the party knows a secret x in $]X-2^\ell, X+2^\ell[$, the signature only guarantees that x lies in the extended interval $]X - 2^{\epsilon(\ell+k)}, X + 2^{\epsilon(\ell+k)}[$.

The security of all these building blocks has been proven in the random oracle model [BR93] under the strong RSA assumption in [CM98b,FO97,FO98]. That is, if $\epsilon > 1$, then the corresponding interactive protocols are statistical (honest-verifier) zero-knowledge proofs of knowledge.

4 The New Group Signature and Identity Escrow Schemes

This section describes our new group signature scheme and tells how an identity escrow scheme can be derived.

As mentioned in Section 2, many recent group signature schemes involve applying two types of non-group signature schemes: one for issuing certificates and one for actual group-signatures, respectively. The security of the former, in particular, is of immediate relevance because it assures, among other things, the coalition-resistance property of a group signature scheme. The reasoning for this assertion is fairly intuitive:

> Each group member obtains a unique certificate from the group manager as part of `JOIN` where each certificate is actually a signature over a secret random message chosen by each member. As a coalition, all group members can be collectively thought of as a single adversary mounting an adaptive chosen message attack consisting of polynomially many instances of `JOIN`.

The main challenge in designing a practical group signature scheme is in finding a signature scheme for the certification of membership that allows the second signature scheme (which is used to produce actual group signatures) to remain efficient. Typically, the second scheme is derived (using the Fiat-Shamir heuristic) from a proof of knowledge of a membership certificate. Hence, the

certification signature scheme must be such that the latter proof can be realized efficiently.

Recall that proving knowledge of a hash function pre-image is, in general, not possible in an efficient manner. Therefore, a candidate signature scheme must replace the hash function with another suitable function. However, because `JOIN` is an interactive protocol between the new member and the group manager, the latter can limit and influence what he signs (e.g., assure that it signs a *random* message).

4.1 The Group Signature Scheme

Let $\epsilon > 1$, k, and ℓ_p be security parameters and let λ_1, λ_2, γ_1, and γ_2 denote lengths satisfying $\lambda_1 > \epsilon(\lambda_2+k)+2$, $\lambda_2 > 4\ell_p$, $\gamma_1 > \epsilon(\gamma_2+k)+2$, and $\gamma_2 > \lambda_1+2$. Define the integral ranges $\Lambda =]2^{\lambda_1} - 2^{\lambda_2}, 2^{\lambda_1} + 2^{\lambda_2}[$ and $\Gamma =]2^{\gamma_1} - 2^{\gamma_2}, 2^{\gamma_1} + 2^{\gamma_2}[$. Finally, let $\mathcal{H}$ be a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^k$. (The parameter ϵ controls the tightness of the statistical zero-knowledgeness and the parameter ℓ_p sets the size of the modulus to use.)

The initial phase involves the group manager (GM) setting the group public and his secret keys: $\mathcal{Y}$ and $\mathcal{S}$.

`SETUP:`

1. Select random secret ℓ_p-bit primes p', q' such that $p = 2p' + 1$ and $q = 2q' + 1$ are prime. Set the modulus $n = pq$.
2. Choose random elements $a, a_0, g, h \in_R \mathrm{QR}(n)$ (of order $p'q'$).
3. Choose a random secret element $x \in_R \mathbb{Z}^*_{p'q'}$ and set $y = g^x \bmod n$.
4. The group public key is: $\mathcal{Y} = (n, a, a_0, y, g, h)$.
5. The corresponding secret key (known only to GM) is: $\mathcal{S} = (p', q', x)$.

Remark 3. The group public key $\mathcal{Y}$ is made available via the usual means (i.e., embedded in some form of a public key certificate signed by a trusted authority). We note that, in practice, components of $\mathcal{Y}$ must be verifiable to prevent framing attacks. In particular, Proposition 1 provides an efficient way to test whether an element has order at least $p'q'$. Then it is sufficient to square this element to make sure it is in $\mathrm{QR}(n)$, with order $p'q'$. GM also needs to provide a proof that n is the product of two safe primes ([CM99a] shows how this can be done).

Suppose now that a new user wants to join the group. We assume that communication between the user and the group manager is secure, i.e., private and authentic. The selection of per-user parameters is done as follows:

`JOIN:`

1. User P_i generates a secret exponent $\tilde{x}_i \in_R]0, n^2[$, a random integer $\tilde{r} \in_R]0, 2^{2\ell_p}[$ and sends $C_1 = g^{\tilde{x}_i} h^{\tilde{r}} \bmod n$ to GM and proves him knowledge of the representation of C_1 w.r.t. bases g and h.
2. GM checks that $C_1 \in \mathrm{QR}(n)$. If this is the case, GM selects α_i and $\beta_i \in_R]0, 2^{\lambda_2}[$ at random and sends (α_i, β_i) to P_i.

3. User P_i computes $x_i = 2^{\lambda_1} + (\alpha_i \tilde{x}_i + \beta_i \bmod 2^{\lambda_2})$ and sends GM the value $C_2 = a^{x_i} \bmod n$. The user also proves to GM:
   (a) that the discrete log of C_2 w.r.t. base a lies in Λ, and
   (b) knowledge of integers u, v, and w such that
      i. u lies in $]-2^{\lambda_2}, 2^{\lambda_2}[$,
      ii. u equals the discrete log of $C_2/a^{2^{\lambda_1}}$ w.r.t. base a, and
      iii. $C_1^{\alpha_i} g^{\beta_i}$ equals $g^u (g^{2^{\lambda_2}})^v h^w$ (see Definition 6).

   (The statements (i–iii) prove that the user's membership secret $x_i = \log_a C_2$ is correctly computed from C_1, α_i, and β_i.)
4. GM checks that $C_2 \in \mathrm{QR}(n)$. If this is the case and all the above proofs were correct, GM selects a random prime $e_i \in_R \Gamma$ and computes $A_i := (C_2\, a_0)^{1/e_i} \bmod n$. Finally, GM sends P_i the new membership certificate $[A_i, e_i]$. (Note that $A_i = (a^{x_i}\, a_0)^{1/e_i} \bmod n$.)
5. User P_i verifies that $a^{x_i} a_0 \equiv {A_i}^{e_i} \pmod n$.

Remark 4. As part of `JOIN`, GM creates a new entry in the membership table and stores $\{[A_i, e_i]$, `JOIN` transcript$\}$ in the new entry. (`JOIN` transcript is formed by the messages received from and sent to the user in the steps above. It is assumed to be signed by the user with some form of a long-term credential.)

Armed with a membership certificate $[A_i, e_i]$, a group member can generate anonymous and unlinkable group signatures on a generic message $m \in \{0,1\}^*$:

`SIGN:`

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

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

2. Randomly choose $r_1 \in_R \pm\{0,1\}^{\epsilon(\gamma_2+k)}$, $r_2 \in_R \pm\{0,1\}^{\epsilon(\lambda_2+k)}$, $r_3 \in_R \pm\{0,1\}^{\epsilon(\gamma_1+2\ell_p+k+1)}$, and $r_4 \in_R \pm\{0,1\}^{\epsilon(2\ell_p+k)}$ and compute:
   (a) $d_1 = {T_1}^{r_1}/(a^{r_2}\, y^{r_3}) \bmod n$, $d_2 = {T_2}^{r_1}/g^{r_3} \bmod n$, $d_3 = g^{r_4} \bmod n$, and $d_4 = g^{r_1}\, h^{r_4} \bmod n$;
   (b) $c = \mathcal{H}(g \| h \| y \| a_0 \| a \| T_1 \| T_2 \| T_3 \| d_1 \| d_2 \| d_3 \| d_4 \| m)$;
   (c) $s_1 = r_1 - c(e_i - 2^{\gamma_1})$, $s_2 = r_2 - c(x_i - 2^{\lambda_1})$, $s_3 = r_3 - c\, e_i\, w$, and $s_4 = r_4 - c\, w$ (all in $\mathbb{Z}$).
3. Output $(c, s_1, s_2, s_3, s_4, T_1, T_2, T_3)$.

A group signature can be regarded as a signature of knowledge of (1) a value $x_i \in \Lambda$ such that $a^{x_i} a_0$ is the value that is ElGamal-encrypted in (T_1, T_2) under y and of (2) an e_i-th root of that encrypted value, where e_i is the first part of the representation of T_3 w.r.t. g and h and that e_i lies in Γ.

A verifier can check the validity of a signature $(c, s_1, s_2, s_3, s_4, T_1, T_2, T_3)$ of the message m as follows.

VERIFY:

1. Compute:

$$c' = \mathcal{H}\big(g\|h\|y\|a_0\|a\|T_1\|T_2\|T_3\|{a_0}^c\,{T_1}^{s_1-c2^{\gamma_1}}/(a^{s_2-c2^{\lambda_1}}\,y^{s_3}) \bmod n\,\|\; {T_2}^{s_1-c2^{\gamma_1}}/g^{s_3} \bmod n\,\|{T_2}^c\,g^{s_4} \bmod n\,\|{T_3}^c\,g^{s_1-c2^{\gamma_1}}\,h^{s_4} \bmod n\,\|\,m\big)\ .$$

2. Accept the signature if and only if $c = c'$, and $s_1 \in \pm\{0,1\}^{\epsilon(\gamma_2+k)+1}$, $s_2 \in \pm\{0,1\}^{\epsilon(\lambda_2+k)+1}$, $s_3 \in \pm\{0,1\}^{\epsilon(\lambda_1+2\ell_p+k+1)+1}$, $s_4 \in \pm\{0,1\}^{\epsilon(2\ell_p+k)+1}$.

In the event that the actual signer must be subsequently identified (e.g., in case of a dispute) GM executes the following procedure:

OPEN:

1. Check the signature's validity via the VERIFY procedure.
2. Recover A_i (and thus the identity of P_i) as $A_i = T_1/{T_2}^x \bmod n$.
3. Prove that $\log_g y = \log_{T_2}(T_1/A_i \bmod n)$ (see Definition 5).

4.2 Deriving an Identity Escrow Scheme

Only minor changes are necessary to construct an identity escrow scheme out of the proposed group signature scheme. Specifically, the SIGN and VERIFY procedures must be replaced by an interactive protocol between a group member (prover) and a verifier. This protocol can be derived from SIGN by replacing the call to the hash function $\mathcal{H}$ by a call to the verifier. That is, the prover sends to the verifier all inputs to the hash function $\mathcal{H}$ and gets back a value $c \in \{0,1\}^{\ell_c}$ randomly chosen by the verifier, with $\ell_c = O(\log \ell_p)$. Then, the prover computes the s_i's and sends these back to the verifier. The verification equation that the verifier uses to check can be derived from the argument to $\mathcal{H}$ in VERIFY. Depending on the choice of the security parameters, the resulting protocol must be repeated sufficiently many times to obtain a small enough probability of error.

5 Related Work

Previously proposed group signature schemes can be divided into two classes: (I) schemes where the sizes of the group public key and/or of group signatures (linearly) depend on the number of group members and (II) schemes where the sizes of the group public key and of group signatures are constant. Most of the early schemes belong to the first class. Although many of those have been proven secure with respect to some standard cryptographic assumption (such as the hardness of computing discrete logarithms) they are inefficient for large groups.

Numerous schemes of Class II have been proposed, however, most are either insecure (or of dubious security) or are grossly inefficient. The only notable and efficient group signature scheme is due to Camenisch and Michels [CM98b].

Our scheme differs from the Camenisch/Michels scheme mainly in the membership certificate format. As a consequence, our `JOIN` protocol has two important advantages:

(1) Our `JOIN` protocol is an order of magnitude more efficient since all proofs that the new group member must provide are efficient proofs of knowledge of discrete logarithms. This is in contrast to the Camenisch/Michels scheme where the group member must prove that some number is the product of two primes. The latter can be realized only with binary challenges.
(2) Our `JOIN` protocol is more secure for the group members, i.e., it is statistical zero-knowledge with respect to the group member's membership secret. The `JOIN` protocol in the Camenisch-Michels scheme is not; in fact, it requires the group member to expose the product of her secret, a prime of special form, and a random prime; such products are in principle susceptible to an attack due to Coppersmith [Cop96]. (Although, the parameters of their scheme can be set such that this attack becomes infeasible.)

Furthermore, the proposed scheme is provably coalition-resistant against an *adaptive* adversary. This offers an extra advantage:

(3) Camenisch and Michels prove their scheme coalition-resistant against a static adversary who is given all certificates as input, whereas our scheme can handle a much more powerful and realistic adversary that is allowed to *adaptively* run the `JOIN` protocol.

6 Security of the Proposed Schemes

In this section we assess the security of the new group signature scheme and the companion escrow identity scheme. We first need to prove that the following theorems hold.

Theorem 1 (Coalition-resistance). *Under the strong RSA assumption, a group certificate* $[A_i = (a^{x_i}\, a_0)^{1/e_i} \bmod n, e_i]$ *with* $x_i \in \Lambda$ *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.*

Proof. Let $\mathcal{M}$ be an attacker that is allowed to adaptively run the `JOIN` and thereby obtain group certificates $[A_j = (a^{x_j}\, a_0)^{1/e_j} \bmod n, e_j]$, $j = 1, \ldots, K$. Our task is now to show that if $\mathcal{M}$ outputs a tuple $\big(\hat{x}; [\hat{A}, \hat{e}]\big)$, with $\hat{x} \in \Lambda$, $\hat{e} \in \Gamma$, $\hat{A} = (a^{\hat{x}}\, a_0)^{1/\hat{e}} \bmod n$, and $(\hat{x}, \hat{e}) \neq (x_j, e_j)$ for all $1 \leq j \leq K$ with non-negligible probability, then the strong RSA assumption does not hold.

Given a pair (n, z), we repeatedly play a random one of the following two games with $\mathcal{M}$ and hope to calculate a pair $(u, e) \in \mathbb{Z}_n^* \times \mathbb{Z}_{>1}$ satisfying $u^e \equiv z \pmod{n}$ from $\mathcal{M}$'s answers. The first game goes as follows:

1. Select $x_1, \ldots, x_K \in \Lambda$ and $e_1, \ldots, e_K \in \Gamma$.
2. Set $a = z^{\prod_{1 \leq l \leq K} e_l} \bmod n$.

3. Choose $r \in_R \Lambda$ and set $a_0 = a^r \bmod n$.
4. For all $1 \le i \le K$, compute $A_i = z^{(x_i+r)\prod_{1\le l\le K; l\neq i} e_l} \bmod n$.
5. Select $g, h \in_R \mathrm{QR}(n)$, $x \in \{1, \ldots, n^2\}$, and set $y = g^x \bmod n$.
6. Run the `JOIN` protocol K times with $\mathcal{M}$ on input (n, a, a_0, y, g, h). Assume we are in protocol run i. Receive the commitment C_1 from $\mathcal{M}$. Use the proof of knowledge of a representation of C_1 with respect to g and h to extract $\tilde{x}_i$ and $\tilde{r}_i$ such that $C_1 = g^{\tilde{x}_i} h^{\tilde{r}_i}$ (this involves rewinding of $\mathcal{M}$). Choose α_i and $\beta_i \in]0, 2^{\lambda_2}[$ such that the prepared $x_i = 2^{\lambda_1} + (\alpha_i \tilde{x}_i + \beta_i \bmod 2^{\lambda_2})$ and send α_i and β_i to $\mathcal{M}$. Run the rest of the protocol as specified until Step 4. Then send $\mathcal{M}$ the membership certificate $[A_i, e_i]$.
   After these K registration protocols are done, $\mathcal{M}$ outputs $\big(\hat{x}; [\hat{A}, \hat{e}]\big)$ with $\hat{x} \in \Lambda$, $\hat{e} \in \Gamma$, and $\hat{A} = (a^{\hat{x}}\, a_0)^{1/\hat{e}} \bmod n$.
7. If $\gcd(\hat{e}, e_j) \neq 1$ for all $1 \le j \le K$ then output $\perp$ and quit. Otherwise, let $\tilde{e} := (\hat{x}+r) \prod_{1\le l\le K} e_l$. (Note that $\hat{A}^{\hat{e}} \equiv z^{\tilde{e}} \pmod n$.) Because $\gcd(\hat{e}, e_j) = 1$ for all $1 \le j \le K$, we have $\gcd(\hat{e}, \tilde{e}) = \gcd(\hat{e}, (\hat{x}+r))$. Hence, by the extended Euclidean algorithm, there exist $\alpha, \beta \in \mathbb{Z}$ s.t. $\alpha\,\hat{e} + \beta\,\tilde{e} = \gcd(\hat{e}, (\hat{x}+r))$. Therefore, letting $u := z^{\alpha}\, \hat{A}^{\beta} \bmod n$ and $e := \hat{e}/\gcd(\hat{e}, (\hat{x}+r)) > 1$ because $\hat{e} > (\hat{x}+r)$, we have $u^e \equiv z \pmod n$. Output (u, e).

The previous game is only successful if $\mathcal{M}$ returns a new certificate $[A(\hat{x}), \hat{e}]$, with $\gcd(\hat{e}, e_j) = 1$ for all $1 \le j \le K$. We now present a game that solves the strong RSA problem in the other case when $\gcd(\hat{e}, e_j) \neq 1$ for some $1 \le j \le K$. (Note that $\gcd(\hat{e}, e_j) \neq 1$ means $\gcd(\hat{e}, e_j) = e_j$ because e_j is prime.)

1. Select $x_1, \ldots, x_K \in \Lambda$ and $e_1, \ldots, e_K \in \Gamma$.
2. Choose $j \in_R \{1, \ldots, K\}$ and set $a = z^{\prod_{1\le l\le K; l\neq j} e_l} \bmod n$.
3. Choose $r \in_R \Lambda$ and set $A_j = a^r \bmod n$ and $a_0 = A_j^{e_j}/a^{x_j} \bmod n$.
4. For all $1 \le i \le K$ $i \neq j$, compute $A_i = z^{(x_i + e_j r - x_j)\prod_{1\le l\le K; l\neq i,j} e_l} \bmod n$.
5. Select $g, h \in_R \mathrm{QR}(n)$, $x \in \{1, \ldots, n^2\}$, and set $y = g^x \bmod n$.
6. Run the `JOIN` protocol K times with $\mathcal{M}$ on input (n, a, a_0, y, g, h). Assume we are in protocol run i. Receive the commitment C_1 from $\mathcal{M}$. Use the proof of knowledge of a representation of C_1 with respect to g and h to extract $\tilde{x}_i$ and $\tilde{r}_i$ such that $C_1 = g^{\tilde{x}_i} h^{\tilde{r}_i} \bmod n$ (this involves rewinding of $\mathcal{M}$). Choose α_i and $\beta_i \in]0, 2^{\lambda_2}[$ such that the prepared $x_i = 2^{\lambda_1} + (\alpha_i \tilde{x}_i + \beta_i \bmod 2^{\lambda_2})$ and send α_i and β_i to $\mathcal{M}$. Run the rest of the protocol as specified until Step 4. Then send $\mathcal{M}$ the membership certificate $[A_i, e_i]$.
   After these K registration protocols are done, $\mathcal{M}$ outputs $\big(\hat{x}; [\hat{A}, \hat{e}]\big)$ with $\hat{x} \in \Lambda$, $\hat{e} \in \Gamma$, and $\hat{A} = (a^{\hat{x}}\, a_0)^{1/\hat{e}} \bmod n$.
7. If $\gcd(\hat{e}, e_j) \neq e_j$ output $\perp$ and quit. Otherwise, we have $\hat{e} = t\, e_j$ for some t and can define $Z := \hat{A}^t/A_j \bmod n$ if $\hat{x} \ge x_j$ and $Z := A_j/\hat{A}^t \bmod n$ otherwise. Hence, $Z \equiv (a^{|\hat{x}-x_j|})^{1/e_j} \equiv (z^{|\tilde{e}|})^{1/e_j} \pmod n$ with $\tilde{e} := (\hat{x} - x_j) \prod_{1\le l\le K; l\neq j} e_l$. Because $\gcd(e_j, \prod_{\substack{1\le l\le K \\ l\neq j}} e_l) = 1$, it follows that $\gcd(e_j, |\tilde{e}|) = \gcd(e_j, |\hat{x} - x_j|)$. Hence, there exist $\alpha, \beta \in \mathbb{Z}$ s.t. $\alpha\, e_j + \beta\, |\tilde{e}| = \gcd(e_j, |\hat{x} - x_j|)$. So, letting $u := z^{\alpha}\, Z^{\beta} \bmod n$ and $e := e_j/\gcd(e_j, |\hat{x} - x_j|) > 1$ because $e_j > |\hat{x} - x_j|$, we have $u^e \equiv z \pmod n$. Output (u, e).

Consequently, by playing randomly one of the Games 1 or 2 until the result is not $\perp$, an attacker getting access to machine $\mathcal{M}$ can solve the strong RSA problem in expected running-time polynomial in K. Because the latter is assumed to be infeasible, it follows that no one but the group manager can generate group certificates. □

Theorem 2. *Under the strong RSA assumption, the interactive protocol underlying the group signature scheme (i.e., the identification protocol of the identity escrow scheme) is a statistical zero-knowledge (honest-verifier) proof of knowledge of a membership certificate and a corresponding membership secret key.*

Proof. The proof that the interactive protocol is statistical zero-knowledge is quite standard. We restrict our attention the proof of knowledge part.

We have to show that the knowledge extractor is able to recover the group certificate once it has found two accepting tuples. Let $(T_1, T_2, T_3, d_1, d_2, d_3, d_4, c, s_1, s_2, s_3, s_4)$ and $(T_1, T_2, T_3, d_1, d_2, d_3, d_4, \tilde{c}, \tilde{s}_1, \tilde{s}_2, \tilde{s}_3, \tilde{s}_4)$ be two accepting tuples. Since $d_3 \equiv g^{s_4} {T_2}^c \equiv g^{\tilde{s}_4} {T_2}^{\tilde{c}} \pmod{n}$, it follows that $g^{s_4 - \tilde{s}_4} \equiv {T_2}^{\tilde{c}-c} \pmod{n}$. Letting $\delta_4 = \gcd(s_4 - \tilde{s}_4, \tilde{c} - c)$, by the extended Euclidean algorithm, there exist $\alpha_4, \beta_4 \in \mathbb{Z}$ s.t. $\alpha_4 (s_4 - \tilde{s}_4) + \beta_4 (\tilde{c} - c) = \delta_4$. Hence,

$$g \equiv g^{(\alpha_4 (s_4 - \tilde{s}_4) + \beta_4 (\tilde{c} - c))/\delta_4} \equiv ({T_2}^{\alpha_4} g^{\beta_4})^{\frac{\tilde{c}-c}{\delta_4}} \pmod{n} \ .$$

Note that we cannot have $\tilde{c} - c < \delta_4$ because otherwise ${T_2}^{\alpha_4} g^{\beta_4}$ is a $(\frac{\tilde{c}-c}{\delta_4})^{\text{th}}$ root of g, which contradicts the strong RSA assumption. Thus, we have $\tilde{c} - c = \delta_4 = \gcd(s_4 - \tilde{s}_4, \tilde{c} - c)$; or equivalently, there exists $\tau_4 \in \mathbb{Z}$ s.t. $s_4 - \tilde{s}_4 = \tau_4(\tilde{c} - c)$. So, because $s_4 + c\,w = \tilde{s}_4 + \tilde{c}\,w$, we have $\tau_4 = w$ and thus obtain

$$A_i = \frac{T_1}{y^{\tau_4}} \bmod n \ .$$

Moreover, because $d_4 \equiv g^{s_1} h^{s_4} (T_3\, g^{-2^{\gamma_1}})^c \equiv g^{\tilde{s}_1} h^{\tilde{s}_4} (T_3\, g^{-2^{\gamma_1}})^{\tilde{c}} \pmod{n}$, we have $g^{s_1 - \tilde{s}_1} \equiv (T_3\, g^{-2^{\gamma_1}})^{\tilde{c}-c} h^{\tilde{s}_4 - s_4} \equiv (T_3\, g^{-2^{\gamma_1}} h^{-\tau_4})^{\tilde{c}-c} \pmod{n}$. Let $\delta_1 = \gcd(s_1 - \tilde{s}_1, \tilde{c} - c)$. By the extended Euclidean algorithm, there exist $\alpha_1, \beta_1 \in \mathbb{Z}$ s.t. $\alpha_1 (s_1 - \tilde{s}_1) + \beta_1 (\tilde{c} - c) = \delta_1$. Therefore, $g \equiv g^{(\alpha_1 (s_1 - \tilde{s}_1) + \beta_1 (\tilde{c} - c))/\delta_1} \equiv [(T_3\, g^{-2^{\gamma_1}} h^{-\tau_4})^{\alpha_1} g^{\beta_1}]^{\frac{\tilde{c}-c}{\delta_1}} \pmod{n}$. This, in turn, implies by the strong RSA assumption that $\tilde{c} - c = \delta_1 = \gcd(s_1 - \tilde{s}_1, \tilde{c} - c)$; or equivalently that there exists $\tau_1 \in \mathbb{Z}$ s.t. $s_1 - \tilde{s}_1 = \tau_1(\tilde{c} - c)$. Consequently, because $s_1 + c(e_i - 2^{\gamma_1}) = \tilde{s} + \tilde{c}(e_i - 2^{\gamma_1})$, we find

$$e_i = 2^{\gamma_1} + \tau_1 \ .$$

Likewise, from $d_2 \equiv {T_2}^{s_1} g^{-s_3} ({T_2}^{-2^{\gamma_1}})^c \equiv {T_2}^{\tilde{s}_1} g^{-\tilde{s}_3} ({T_2}^{-2^{\gamma_1}})^{\tilde{c}} \pmod{n}$, it follows that $g^{s_3 - \tilde{s}_3} \equiv ({T_2}^{\tau_1 + 2^{\gamma_1}})^{\tilde{c}-c} \pmod{n}$. Therefore, by the extended Euclidean algorithm, we can conclude that there exists $\tau_3 \in \mathbb{Z}$ s.t. $s_3 - \tilde{s}_3 = \tau_3(\tilde{c} - c)$. Finally, from $d_1 \equiv {T_1}^{s_1} a^{-s_2} y^{-s_3} ({T_1}^{-2^{\gamma_1}} a^{2^{\lambda_1}} a_0)^c \equiv {T_1}^{\tilde{s}_1} a^{-\tilde{s}_2} y^{-\tilde{s}_3} ({T_1}^{-2^{\gamma_1}} a^{2^{\lambda_1}} a_0)^{\tilde{c}} \pmod{n}$, we obtain $a^{\tilde{s}_2 - s_2} \equiv ({T_1}^{\tau_1 + 2^{\gamma_1}} y^{-\tau_3} a^{-2^{\lambda_1}} {a_0}^{-1})^{\tilde{c}-c} \pmod{n}$ and similarly conclude that there exists $\tau_2 \in \mathbb{Z}$ s.t. $s_2 - \tilde{s}_2 = \tau_2(\tilde{c} - c)$. Because $s_2 + c(x_i - 2^{\lambda_1}) = \tilde{s}_2 + \tilde{c}(x_i - 2^{\lambda_1})$, we recover

$$x_i = 2^{\lambda_1} + \tau_2 \,,$$

which concludes the proof. □

Corollary 2. *The* `JOIN` *protocol is zero-knowledge w.r.t. the group manager. Furthermore, the user's membership secret key x_i is a random integer from Λ.*

Proof. Straight-forward. □

Corollary 3. *In the random oracle model the group signature scheme presented in Section 4 is secure under the strong RSA and the decisional Diffie-Hellman assumption.*

Proof. We have to show that our scheme satisfies all the security properties listed in Definition 1.

Correctness: By inspection.

Unforgeability: Only group members are able to sign messages on behalf of the group: This is an immediate consequence of Theorem 2 and the random oracle model, that is, if we assume the hash function $\mathcal{H}$ behaves as a random function.

Anonymity: Given a valid signature $(c, s_1, s_2, s_3, s_4, T_1, T_2, T_3)$ identifying the actual signer is computationally hard for everyone but the group manager: Because of Theorem 2 the underlying interactive protocol is statistically zero-knowledge, no information is statistically revealed by (c, s_1, s_2, s_3, s_4) in the random oracle model. Deciding whether some group member with certificate $[A_i, e_i]$ originated requires deciding whether the three discrete logarithms $\log_y T_1/A_i$, $\log_g T_2$, and $\log_g T_3/g^{e_i}$ are equal. This is assumed to be infeasible under the decisional Diffie-Hellman assumption and hence anonymity is guaranteed.

Unlinkability: Deciding if two signatures $(T_1, T_2, T_3, c, s_1, s_2, s_3, s_4)$ and $(\tilde{T}_1, \tilde{T}_2, \tilde{T}_3, \tilde{c}, \tilde{s}_1, \tilde{s}_2, \tilde{s}_3, \tilde{s}_4)$ were computed by the same group member is computationally hard. Simiarly as for Anonymity, the problem of linking two signatures reduces to decide whether the three discrete logarithms $\log_y T_1/\tilde{T}_i$, $\log_g T_2/\tilde{T}_2$, and $\log_g T_3/\tilde{T}_3$ are equal. This is, however, impossible under Decisional Diffie-Hellman Assumption.

Exculpability: Neither a group member nor the group manager can sign on behalf of other group members: First note that due to Corollary 2, GM does not get any information about a user's secret x_i apart from a^{x_i}. Thus, the value x_i is computationally hidden from GM. Next note that T_1, T_2, and T_3 are an unconditionally binding commitments to A_i and e_i. One can show that, if the factorization of n would be publicly known, the interactive proof underlying the group signature scheme is a proof of knowledge of the discrete log of $A_i^{e_i}/a_0$ (provided that ℓ_p is larger than twice to output length of the hash function / size of the challenges). Hence, not even the group manager can sign on behalf of P_i because computing discrete logarithms is assumed to be infeasible.

Traceability: The group manager is able to open any valid group signature and *provably* identify the actual signer: Assuming that the signature is valid, this implies that T_1 and T_2 are of the required form and so A_i can be uniquely recovered. Due to Theorem 1 a group certificate $[A_i = A(x_i), e_i]$ with $x_i \in \Lambda$

and $e_i \in \Gamma$ can only be obtained from via the `JOIN` protocol. Hence, the A_i recovered can be uniquely be linked to an instance of the `JOIN` protocol and thus the user P_i who originated the signature can be identified.

Coalition-resistance: Assuming the random oracle model, this follows from Theorems 1 and 2.

□

Corollary 4. *The identity escrow scheme derived from our group signature scheme is secure under the strong RSA and the decisional Diffie-Hellman assumption.*

Proof. The proof is essentially the same as for Corollary 3, the difference being that we do not need the random oracle model but can apply Theorems 1 and 2 directly. □

7 Conclusions

This paper presents a very efficient and provably secure group signature scheme and a companion identity escrow scheme that are based on the strong RSA assumption. Their performance and security appear to significantly surpass those of prior art. Extending the scheme to a blind group-signature scheme or to split the group manager into a membership manager and a revocation manager is straight-forward (cf. [CM98a,LR98]).

References

[ASW98] N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures. In *Advances in Cryptology – EUROCRYPT '98*, volume 1403 of *LNCS*, pages 591–606, Springer-Verlag, 1998.

[Ate99] G. Ateniese. Efficient verifiable encryption (and fair exchange) of digital signatures. In *6st ACM Conference on Computer and Communication Security*, ACM Press, 1999.

[BF97] N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology — EUROCRYPT '97*, vol. 1233 of *LNCS*, pp. 480–494, Springer-Verlag, 1997.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *1st ACM Conference on Computer and Communication Security*, pp. 62–73, ACM Press, 1993.

[Bon98] D. Boneh. The decision Diffie-Hellman problem. In *Algorithmic Number Theory (ANTS-III)*, vol. 1423 of *LNCS*, pp. 48–63, Springer-Verlag, 1998.

[Bra93] S. Brands. An efficient off-line electronic cash system based on the representation problem. Technical Report CS-R9323, Centrum voor Wiskunde en Informatica, April 1993.

[CD98] J. Camenisch and I. Damgård. Verifiable encryption and applications to group signatures and signature sharing. BRICS Technical Report, RS-98-32.

[CM98a] J. Camenisch and M. Michels. A group signature scheme with improved efficiency. In *Advances in Cryptology — ASIACRYPT '98*, vol. 1514 of *LNCS*, pp. 160–174, Springer-Verlag, 1998.

[CM98b] ———. A group signature scheme based on an RSA-variant. Technical Report RS-98-27, BRICS, University of Aarhus, November 1998. An earlier version appears in [CM98a].

[CM99a] ———. Proving in zero-knowledge that a number is the product of two safe primes. In *Advances in Cryptology — EUROCRYPT '99*, vol. 1592 of *LNCS*, pp. 107–122, Springer-Verlag, 1999.

[CM99b] ———. Separability and efficiency for generic group signature schemes. In *Advances in Cryptology — CRYPTO '99*, vol. 1666 of *LNCS*, pp. 413–430, Springer-Verlag, 1999.

[CP95] L. Chen and T. P. Pedersen. New group signature schemes. In *Advances in Cryptology — EUROCRYPT '94*, vol. 950 of *LNCS*, pp. 171–181, 1995.

[CS97] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In *Advances in Cryptology — CRYPTO '97*, vol. 1296 of *LNCS*, pp. 410–424, Springer-Verlag, 1997.

[Cam98] J. Camenisch. Group signature schemes and payment systems based on the discrete logarithm problem. PhD thesis, vol. 2 of *ETH Series in Information Security an Cryptography*, Hartung-Gorre Verlag, Konstanz, 1998. ISBN 3-89649-286-1.

[Cop96] D. Coppersmith. Finding a small root of a bivariatre interger equation; factoring with high bits known. In *Advances in Cryptology — EUROCRYPT '96*, volume 1070 of *LNCS*, pages 178–189. Springer Verlag, 1996.

[CvH91] D. Chaum and E. van Heyst. Group signatures. In *Advances in Cryptology — EUROCRYPT '91*, vol. 547 of *LNCS*, pp. 257–265, Springer-Verlag, 1991.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6): 644–654, 1976.

[FO97] E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology — CRYPTO '97*, vol. 1297 of *LNCS*, pp. 16–30, Springer-Verlag, 1997.

[FO98] ———. A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In *Advances in Cryptology — EUROCRYPT '98*, vol. 1403 of *LNCS*, pp. 32–46, Springer-Verlag, 1998.

[FS87] A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. In *Advances in Cryptology — CRYPTO '86*, vol. 263 of *LNCS*, pp. 186–194, Springer-Verlag, 1987.

[GMR88] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[KP98] J. Kilian and E. Petrank. Identity escrow. In *Advances in Cryptology — CRYPTO '98*, vol. 1642 of *LNCS*, pp. 169–185, Springer-Verlag, 1998.

[LR98] A. Lysyanskaya and Z. Ramzan. Group blind digital signatures: A scalable solution to electronic cash. In *Financial Cryptography (FC '98)*, vol. 1465 of *LNCS*, pp. 184–197, Springer-Verlag, 1998.

[Sch91] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.