Group Undeniable Signatures

YUH-DAUH LYUU
Dept. of Computer Science & Information Engineering and Dept. of Finance
National Taiwan University
No 1, Sec 4, Roosevelt Rd, Taipei, Taiwan
lyuu@csie.ntu.edu.tw

MING-LUEN WU
Dept. of Computer Science & Information Engineering
National Taiwan University
No 1, Sec 4, Roosevelt Rd, Taipei, Taiwan
d5526009@csie.ntu.edu.tw

Abstract: - A group undeniable signature scheme is proposed in which each group member can sign on behalf of the group without revealing his identity and the verification of a signature can only be done by interaction with the group manager. For business applications, group undeniable signatures can be used when the signatures are commercially valuable to competitors. If a group member is falsely accused of having signed a signature, the group manager has the ability to prove his innocence. In case of later disputes, the group manager can track down which member signed the signature. Our scheme can be proven to be unforgeable, signature-simulatable and coalition-resistant. The confirmation and denial protocols are also zero-knowledge. Furthermore, the time, space and communication complexity are independent of the group size.

Key-Words: - Group signature, Undeniable signature, Signature of knowledge, Unforgeability, Coalition resistance

1 Introduction

Digital signatures are bonded with messages and signers such that everyone can verify whether the message really comes from the alleged signer. Generally, a signer uses a secret value to generate the signature and publishes the corresponding public information for universal verification. However, universal verifiability might not suit the circumstances when the ability to verify signatures can be used against the signers' interests. For example, a competitor may inquire about prices and request the merchant to sign the message. If anyone can verify the signature, the merchant's power to give differential quotes to clients of different standing may be compromised. Limiting the ability to verify signatures is hence desirable. Chaum and van Antwerpen [4] initiate an undeniable signature scheme in which interaction with the signer is needed to verify a signature and the signer can disavow an invalid signature through a denial protocol. Non-repudiation requires that the signer cannot deny his signature unless the signature is indeed invalid.

A group signature scheme allows a group member to sign messages on behalf of the group without revealing his identity. Nevertheless, in case of later disputes, a designated group manager can open the signature, thus tracing the signer. At the same time, anyone—including the group manager—cannot misattribute a valid signature. The concept of group signature is first introduced by Chaum and van Heyst [5], and Camenisch and Stadler [2] present the first scheme in which the size of the public key and signatures is independent of the group size. Analogous to standard digital signatures, group signatures are non-repudiatable and universally verifiable.

In this paper, we introduce a new concept, group undeniable signature. A group undeniable signature is like an ordinary group signature except that verifying signatures must involve the group manager. The notion of group undeniable signature combines group signatures and undeniable signatures. Applications of group undeniable signatures include validating price lists, press releases, and digital contracts when the signatures for companies are commercially valuable to competitors. Our scheme is based on signatures of knowledge [2] and undeniable signature schemes [3]. The proposed scheme is existentially unforgeable against adaptive chosen message attacks. It is also signature-simulatable and coalition-resistant under reasonable number-theoretic complexity assumptions and in the random oracle model [1]. The signature confirmation and denial protocols can be zero-knowledge by applying the commitment techniques.

2 Model

A group undeniable signature scheme consists of six components.

System setup: The group's secret and public keys are generated for the group manager.

Join: To become a group member, a person first generates his secret key and membership key, and then registers the membership key with the group manager. Afterwards the group manager sends him the membership certificate.

Sign: A group member signs messages using his secret key, his membership certificate, and the group public key.

Signature confirmation protocol: To verify a signature requires interacting with the group manager.

Signature denial protocol: The group manager can prove to anyone that an invalid signature is indeed invalid through a signature denial protocol.

Open: The group manager can trace the identity of the member who signs a given message.

In general, a group undeniable signature scheme should satisfy the following security considerations.

Unforgeability: Only a group member can sign on behalf of the group.

Unlinkability: No one except the group manager can tell whether two different signatures are generated by the same group member.

Anonymity: No one except the group manager can identify the signer.

Non-transferability: Only the group manager can prove the validity or invalidity of signatures.

Zero knowledge: The confirmation and denial protocols reveal no extra information beyond the validity or invalidity of signatures.

Exculpability: Neither the group manager nor a group member can sign on behalf of another group member.

Traceability. The group manager can identify the signer of a valid signature.

Coalition-resistance: A colluding subset of group members can not generate valid signatures that can not be traced by the group manager.

The efficiency of a group undeniable signature scheme involves the following parameters of interest.

- The size of the group signature.
- The size of the group public key.
- The efficiency of System setup, Join and Open.
- The efficiency of Sign and Verify (including the confirmation and deniable protocols).

3 Number-theoretic Preliminaries

For positive integer n, $\mathbb{Z}_n$ denotes the ring of integers modulo n, and $\mathbb{Z}_n^*$ denotes the multiplicative group modulo n. Let $\phi(n)$ denote Euler's phi function, which gives the number of positive integers $m \in \{1, 2, \ldots, n-1\}$ such that $\gcd(m, n) = 1$. Expression "$r \in_R I''$ means that r is chosen randomly from set I. The least positive integer d such that $g^d \equiv 1 \pmod{M}$ is called the *order* of g modulo M, and is denoted by $\mathrm{ord}_M g$ or simply $\mathrm{ord}(g)$ if M is understood.

Fact 3.1. *Let $G = \langle g \rangle$ be a cyclic group generated by g. If ord$(g) = n$ and if r is a positive integer, then ord$(g^r) = n/\gcd(n, r)$.*

Let $G = \langle g \rangle$ be the cyclic group generated by g with order n. The following problem is assumed to be intractable whether n is known or not.

Equality of Discrete Logarithms (EDL): Given $x, y \in_R G = \langle f \rangle = \langle g \rangle$, determine the equality of $\log_f x$ and $\log_g y$ over $\mathbb{Z}_n$.

4 Signatures of Knowledge

Signatures of knowledge allow a prover to prove the knowledge of a secret with respect to some public information noninteractively. In this section, we review the important signatures of knowledge to be employed as building blocks of our signature scheme.

Let G be a cyclic group generated by g with order n, where n is the product of two large primes. We denote by Greek letters the elements whose knowledge is to be proven and by all other letters the elements that are publicly known. Denote by $\|$ the concatenation of two binary strings and by $\wedge$ the logical conjunction. A hash function H is coalition-resistant if it is infeasible to find two different inputs x and y such that $H(x) = H(y)$. Assume $\mathcal{H}$ is a collision-resistant hash function throughout the paper.

Knowledge of a representation. Let $y_1 = \prod_{j=1}^{\ell_1} g_{b_{1j}}^{x_{e_{1j}}}, \ldots, y_w = \prod_{j=1}^{\ell_w} g_{b_{wj}}^{x_{e_{wj}}}$, where $e_{ij} \in \{1, \ldots, u\}$ and $b_{ij} \in \{1, \ldots, k\}$. A signature of knowledge of a representation $(x_1, \ldots, x_u)$ of $y_1, \ldots, y_w$ with respect to generators $g_1, \ldots, g_k$ on message m is $(c, s_1, s_2, \ldots, s_u)$. It can be generated as follows. Choose $r_i \in_R \mathbb{Z}_n$ for $i = 1, \ldots, u$ and then compute $c = \mathcal{H}(m \parallel y_1 \parallel \ldots \parallel y_w \parallel g_1 \parallel \ldots \parallel g_k \parallel \{\{e_{ij}, b_{ij}\}_{j=1}^{\ell_i}\}_{i=1}^{w} \parallel \prod_{j=1}^{\ell_1} g_{b_{1j}}^{r_{e_{1j}}} \parallel \cdots \parallel \prod_{j=1}^{\ell_w} g_{b_{wj}}^{r_{e_{wj}}})$ and

$s_i = r_i - cx_i \bmod n$, for $i = 1, \ldots, u$. Such a signature can be computed by a signer who knows the representation $(x_1, \ldots, x_u)$. We denote this signature by

$$\text{SKREP}\left[(\alpha_1, \ldots, \alpha_u) : (y_1 = \prod_{j=1}^{\ell_1} g_{b_{1j}}^{\alpha_{e_{1j}}}) \wedge \cdots \wedge (y_w = \prod_{j=1}^{\ell_w} g_{b_{wj}}^{\alpha_{e_{wj}}})\right](m).$$

Anyone can verify the signature by checking whether $c = \mathcal{H}(m \parallel y_1 \parallel \ldots \parallel g_k \parallel \{\{e_{ij}, b_{ij}\}_{j=1}^{\ell_i}\}_{i=1}^{w} \parallel \prod_{j=1}^{\ell_1} g_{b_{1j}}^{s_{e_{1j}}} y_1^c \parallel \cdots \parallel \prod_{j=1}^{\ell_w} g_{b_{wj}}^{s_{e_{wj}}} y_w^c)$.

Knowledge of roots of representations. Such a signature is used to prove that one knows the e-th root x of the g-part of a representation of $v = f^w g^{x^e} \in G = \langle f \rangle = \langle g \rangle$. A signature of knowledge of the pair (w, x) of $v = f^w g^{x^e}$ on message m consists of two components.

- $(v_1, \ldots, v_{e-1})$, where $v_i = f^{r_i} g^{x^i}$ and $r_i \in_R \mathbb{Z}_n$ for $i = 1, \ldots, e-1$.
- SKREP$[(\gamma_1, \gamma_2 \ldots, \gamma_e, \delta) : v_1 = f^{\gamma_1} g^{\delta} \wedge v_2 = f^{\gamma_2} v_1^{\delta} \wedge \cdots \wedge v_{e-1} = f^{\gamma_{e-1}} v_{e-2}^{\delta} \wedge v = f^{\gamma_e} v_{e-1}^{\delta}](m)$.

We denote the complete signature by SKRREP$[(\alpha, \beta) : v = f^{\alpha} g^{\beta^e}](m)$. If a small integer e is chosen, the signature can be generated efficiently. A signer who knows (w, x) can generate such a signature. The first component is computed directly. Because $r_i \in_R \mathbb{Z}_n$, we know $v_i \in_R G$. Furthermore, because of equations $v_i = f^{r_i} g^{x^i}$ and $v = f^w g^{x^e}$, we let $\gamma_1 = r_1, \gamma_i = r_i - xr_{i-1}$ for $i = 2, \ldots, e-1$, $\gamma_e = w - xr_{e-1}$, and $\delta = x$. Hence, the second component can be obtained.

Knowledge of roots of discrete logarithms. Assume f is another generator of $G = \langle g \rangle$ and $\log_g f$ is not known. A signature of knowledge of the e-th root x of the discrete logarithm of $y = g^{x^e}$ on the message m comprises two components.

- SKRREP$[(\alpha, \beta) : y = f^{\alpha} g^{\beta^e}](m)$.
- SKREP$[\gamma : y = g^{\gamma}](m)$.

We denote the whole signature by SKRDL$[\alpha : y = g^{\alpha^e}](m)$. With the secret x, the signer knows a representation $(0, x^e)$ of $y = f^0 g^{x^e}$ to generators f and g. This must be the only representation the signer knows; otherwise, he would be able to compute $\log_g f$. This implies $\alpha = 0, \beta = x$, and $\gamma = x^e$, and the two underlying signatures can be computed. To verify such a signature, one must check the correctness of the two components.

According to results in [6, Section 3], in the random oracle model, we can derive that the above signatures are simulatable and existentially unforgeable against adaptive chosen message attacks under the related number-theoretic complexity assumptions.

5 The Scheme

5.1 System Setup

To derive the group secret key and the group public key, the group manager computes the following values.

- An RSA public key $(n = p_1 p_2, e_R)$ and secret key d_R.
- A cyclic group $G = \langle g \rangle$ of order n.
- $f = g^a, S_g = g^b, u = g^h, t = u^{\rho}$ where a, b, h, and $\rho \in_R \mathbb{Z}_n^*$.
- (e, d) for $e, d \in \mathbb{Z}_n^*$ such that $ed \equiv 1 \pmod{n}$.
- $S_f = f^d$.

Note that n must be chosen such that factoring n and solving discrete logarithm in G are intractable. Here is one way to pick $G = \langle g \rangle$. Let g_0 be a generator of $\mathbb{Z}_p^*$, a cyclic group, where p is a prime. If we let $G = \langle g_0^{(p-1)/n} \rangle$ and $n \mid (p-1)$, then G is a subgroup of $\mathbb{Z}_p^*$. By Fact 3.1, $g = g_0^{(p-1)/n}$ has order n and is hence the desired generator of G. The orders of f, S_f, S_g, u, and t are also n. The group manager keeps $(b, d, d_R, e, \rho^{-1}, p_1, p_2)$ as the group secret key and opens $(n, e_R, f, g, S_f, S_g, u, t)$ as the group public key.

5.2 Join

When Alice wants to join the group, she chooses the secret key $y \in_R \mathbb{Z}_n^*$ to compute her membership key $z = g^y$. Then Alice sends z to the group manager and proves to the group manager that she knows the discrete logarithm of z without revealing it. Next, the group manager chooses $c \in_R \mathbb{Z}_n^*$ such that $(zg^c)^{p_1} \neq 1$ and $(zg^c)^{p_2} \neq 1$ (this is doable by testing at most three continuous integers). Note that $\gcd(y + c, n) = 1$. Then the group manager computes Alice's membership certificate (x, v, w) and sends it to Alice, where $x = g^c, v = (c + b)^{d_R} \bmod n$, and $w = (zx)^d$. The 4-tuple (y, x, v, w) is called a valid signing key. The group manager must choose distinct c's for different members and must prevent anyone from knowing c's. Fact 3.1 implies that $\text{ord}(z) = \text{ord}(x) = \text{ord}(w) = n$.

5.3 Sign

Given a message m, Alice computes the following values.

- $\hat{g} = g^r$ for $r \in_R \mathbb{Z}_n^*$ (note that $G = \langle \hat{g} \rangle$).
- $Z_0 = S_g^r$, $Z_1 = \hat{g}^y$, $Z_2 = x^r$, $A_1 = g^y u^r$, $A_2 = t^r$.
- $S_0 = \mathrm{SKREP}[(\alpha, \beta) : \hat{g} = g^\beta \wedge Z_0 = S_g^\beta \wedge Z_1 = \hat{g}^\alpha \wedge A_1 = g^\alpha u^\beta \wedge A_2 = t^\beta](m)$.
- $S_1 = \mathrm{SKRDL}[\gamma : Z_2 Z_0 = \hat{g}^{\gamma^{e_R}}](m)$.
- $S_2 = w^r$.

Alice's group undeniable signature on m is $S = (\hat{g}, Z_0, Z_1, Z_2, A_1, A_2, S_0, S_1, S_2)$. We call S a valid group undeniable signaure if S is generated using a valid signing key. The correctness of S is based on the correctness of S_0, S_1, and S_2.

5.4 Signature Confirmation Protocol

A signature confirmation protocol is an interactive protocol between the group manager and a verifier whereby the group manager can convince the verifier that the signature is valid. However, the group manager cannot cheat the verifier into accepting an invalid signature as valid except with a very small probability. In the following, we denote by $\mathcal{P}$ the group manager and by $\mathcal{V}$ the verifier. The notation $X \longrightarrow Y : Z$ represents that X sends Z to Y. In the confirmation protocol, common inputs to $\mathcal{P}$ and $\mathcal{V}$ include the message m, the group public key and the alleged signature S. The secret input to $\mathcal{P}$ is the group secret key.

To be convinced that S is valid, first $\mathcal{V}$ checks S_0 and S_1. If either is incorrect, then $\mathcal{V}$ recognizes that S is invalid. Otherwise, $\mathcal{P}$ and $\mathcal{V}$ perform the following steps:

1. $\mathcal{V} \longrightarrow \mathcal{P} : A$
   $\mathcal{V}$ chooses $e_1, e_2 \in_R \mathbb{Z}_n^*$ and computes $A = S_2^{e_1} S_f^{e_2}$.
2. $\mathcal{P} \longrightarrow \mathcal{V} : B$
   $\mathcal{P}$ computes $B = A^e$.
3. $\mathcal{V}$ verifies that $(Z_1 Z_2)^{e_1} f^{e_2} = B$.
   If the equality holds then $\mathcal{V}$ accepts S as a valid signature for m. Otherwise $\mathcal{V}$ cannot determine S is valid or invalid.

The following theorem says that $\mathcal{V}$ accepts valid signatures.

Theorem 5.1. *If S is a valid group undeniable signature, then the verifier will accept S as a valid signature for m.*

Proof. Obviously, S_0 and S_1 must be correct. Furthermore, because $w = (g^{y+c})^d$, we have $S_2 = w^r = ((g^{y+c})^d)^r = ((\hat{g})^{y+c})^d = (Z_1 Z_2)^d$. So $B = A^e = ((S_2)^{e_1} (S_f)^{e_2})^e = (Z_1 Z_2)^{e_1} f^{e_2}$. □

Next we show that $\mathcal{P}$ cannot cheat $\mathcal{V}$ into accepting invalid signatures as valid except with a very small probability.

Theorem 5.2. *If S is not a valid group undeniable signature, then a verifier will accept S as a valid signature for m with probability $1/n$.*

Proof. If S_0 or S_1 is incorrect, a verifier recognizes S as invalid. Now suppose S_0 and S_1 are correct. Because S is generated without a valid signing key, $S_2 \neq (Z_1 Z_2)^d$. $\mathcal{P}$ can make $\mathcal{V}$ accept the signature only if $\mathcal{P}$ can find $B = (Z_1 Z_2)^{e_1} f^{e_2}$ such that (e_1, e_2) satisfies $A = S_2^{e_1} S_f^{e_2}$. That is, (e_1, e_2) satisfies the following two equations:

$$A = S_2^{e_1} S_f^{e_2} \tag{1}$$

$$B = (Z_1 Z_2)^{e_1} f^{e_2}, \tag{2}$$

where $S_2 \neq (Z_1 Z_2)^d$. As the order of f is n, we let $A = f^i, B = f^j, S_2 = f^k$, and $Z_1 Z_2 = f^\ell$ for some $i, j, k, \ell \in Z_n$. Recall $S_f = f^d$. From (1) and (2), we have $i = k e_1 + d e_2 \bmod n$ and $j = \ell e_1 + e_2 \bmod n$. As $f^k \neq f^{\ell d}, k \neq \ell d \pmod n$ and there is a unique solution for (e_1, e_2).
By Fact 3.1, the orders of S_2, S_f, and $Z_1 Z_2$ are all n; hence there are n ordered pairs (e_1, e_2) satisfying $A = S_2^{e_1} S_f^{e_2}$. $\mathcal{P}$ cannot identify which among them was used to compute A by $\mathcal{V}$. In addition, every B is a correct response for exactly one of the possible ordered pairs. Consequently, the probability that $\mathcal{P}$ will give $\mathcal{V}$ the correct B is $1/n$. □

To illustrate the protocol clearly, the above steps omit the zero-knowledge part. We can make the protocol zero-knowledge by modifying Step 2 as follows: $\mathcal{P}$ commits B to $\mathcal{V}$ using a commitment scheme such that $\mathcal{V}$ cannot learn what B is unless $\mathcal{V}$ sends the correct e_1 and e_2 to $\mathcal{P}$. Because $B = (Z_1 Z_2)^{e_1} f^{e_2}$ can be computed using the correct e_1 and e_2, $\mathcal{P}$ reveals no extra information to $\mathcal{V}$. Accordingly, the whole protocol is zero-knowledge.

5.5 Signature Denial Protocol

A signature denial protocol allows $\mathcal{P}$ to convince $\mathcal{V}$ of the fact that an invalid signature is indeed invalid. However, $\mathcal{P}$ cannot make $\mathcal{V}$ believe that a valid signature is invalid except with a very small probability. In the denial protocol, the common inputs to $\mathcal{P}$ and $\mathcal{V}$ include two constants c_1 and c_2, the message m, the group public key, and the alleged signature S. The secret input to $\mathcal{P}$ is the group secret key.

We first present how $\mathcal{P}$ can make $\mathcal{V}$ reject an invalid signature S. $\mathcal{V}$ starts by checking S_0 and S_1. If either is incorrect, then $\mathcal{V}$ recognizes that S is invalid. Otherwise, $\mathcal{P}$ and $\mathcal{V}$ repeat the following steps c_2 times.

1. $\mathcal{V} \longrightarrow \mathcal{P} : A_1, A_2$
   $\mathcal{V}$ chooses $e_1 \in_R \mathbb{Z}_{c_1}, e_2 \in_R \mathbb{Z}_n$ and computes $A_1 = (Z_1Z_2)^{e_1} f^{e_2}$, $A_2 = S_2^{e_1} S_f^{e_2}$.
2. $\mathcal{P} \longrightarrow \mathcal{V} : B$
   $\mathcal{P}$ finds e_1 such that $A_1/A_2^e = (Z_1Z_2/S_2^e)^{e_1}$ and computes $B = e_1$.
3. $\mathcal{V}$ checks whether $B = e_1$.
   If the equality holds, then $\mathcal{V}$ is convinced that S is invalid.

If $\mathcal{V}$ is convinced of S's invalidity c_2 times, S is rejected for invalidity. It is noteworthy that $\mathcal{P}$ performs at most c_1c_2 operations to find the correct e_1's.

The following theorem says that $\mathcal{P}$ can convince $\mathcal{V}$ of the fact that an invalid signature is indeed invalid.

Theorem 5.3. *If S is not a valid group undeniable signature, then a verifier will accept S as an invalid signature for m.*

Proof. If S_0 or S_1 is incorrect, a verifier will recognize S as an invalid signature. Suppose S_0 and S_1 are both correct. Because S is generated without a valid signing key, $S_2 \neq (Z_1Z_2)^d$ and therefore $S_2^e \neq Z_1Z_2$. As $A_1/A_2^e = (Z_1Z_2/S_2^e)^{e_1}$, $\mathcal{P}$ can always find the required e_1. This implies that $\mathcal{V}$ will reject S for invalidity. □

Next we prove that $\mathcal{P}$ cannot fool $\mathcal{V}$ into accepting a valid signature as an invalid signature except with a small probability.

Theorem 5.4. *If S is a valid group undeniable signature, then a verifier will accept S as an invalid signature for m with probability $1/c_1^{c_2}$.*

Proof. Because S is valid, S_0 and S_1 are correct and $S_2 = (Z_1Z_2)^d$. Therefore $S_2^e = Z_1Z_2$. We have $A_1/A_2^e = (Z_1Z_2/S_2^e)^{e_1} = 1$. In this case $\mathcal{P}$ can only randomly choose e_1 from $\mathbb{Z}_{c_1}$. Consequently, $\mathcal{V}$ will accept S as an invalid signature for m with probability $1/c_1^{c_2}$. □

To illustrate this protocol clearly, we omit the zero-knowledge part. Applying a commitment scheme, we can make the protocol zero-knowledge by modifying Step 2 as follows: $\mathcal{P}$ commits B to $\mathcal{V}$ such that $\mathcal{V}$ cannot learn what B is unless $\mathcal{V}$ sends the correct e_2 to $\mathcal{P}$. The correct e_2 means that e_2 satisfies $A_1 = (Z_1Z_2)^{e_1} f^{e_2}$ and $A_2 = S_2^{e_1} S_f^{e_2}$, where e_1 is the value found by $\mathcal{P}$. This can be checked by $\mathcal{P}$. Because the correct e_2 ensures that $\mathcal{P}$ and $\mathcal{V}$ have the same e_1, $\mathcal{P}$ reveals no extra information to $\mathcal{V}$. Accordingly, the whole protocol is zero-knowledge.

5.6 Open

Given a valid signature S, the group manager can compute $z = A_1 A_2^{-(\rho^{-1} \bmod n)}$. The signer with the membership key z can be traced directly. We notice that z_P is an ElGamal decryption of (A_1, A_2) with respect to the secret key ρ^{-1} mod n.

6 Security Analysis

Exculpability. Because the discrete logarithm problem is intractable, neither the group manager nor a group member can compute the secret key y of another group member. Thus, it is infeasible to frame another member.
Unforgeability. Recall that any valid signature S must contain correct S_0, S_1, and S_2. Considering S_2, an attacker must obtain $S_2 = \xi^d$, where $\xi = \xi_1\xi_2$ with $\xi_1 = \hat{g}^y$ and $\xi_2 Z_0 = \hat{g}^{v^{e_R}}$. Using adaptive chosen message attacks, the attacker can compute many (ξ, ξ^d)'s with random ξ's, but he cannot learn d. From a random ξ, the two values ξ_1 and ξ_2 must be computed such that S_0 and S_1 are correct. Here $S_0 =$SKREP$[(\alpha, \beta) : \hat{g} = g^\beta \wedge Z_0 = S_g^\beta \wedge \xi_1 = \hat{g}^\alpha \wedge A_1 = g^\alpha u^\beta \wedge A_2 = t^\beta](m)$ and $S_1 =$SKRDL$[\gamma : \xi_2 Z_0 = \hat{g}^{\gamma^{e_R}}](m)$. Next we show that the attacker cannot simultaneously obtain correct S_0, S_1 and S_2. Note that the attacker cannot compute S_0 and S_1 without knowing α and γ, respectively. Now, to obtain S_0 from a (ξ, ξ^d), the attacker chooses y and has $\xi_1 = \hat{g}^y$. So $\xi_2 = \xi\xi_1^{-1}$. Assume $\xi_2 = \hat{g}^c$. Because the value $v = (c+b)^{d_R}$ mod n satisfying $\xi_2 Z_0 = \hat{g}^{v^{e_R}}$ cannot be obtained, S_1 is existentially unforgeable against adaptive chosen message attacks. Consequently, we have the following theorem.

Theorem 6.1. *Our signature scheme is existentially unforgeable against adaptive chosen message attacks.*

Unlinkability, Anonymity, Non-transferability. These properties hold if the signatures are simulatable. The signatures can be simulated as follows. Let $S = (\hat{g}, Z_0, Z_1, Z_2, A_1, A_2, S_0, S_1, S_2)$ be a valid signature. Assume the signer's membership key z equals u^{r_z} for some $r_z \in \mathbb{Z}_n^*$. So $A_1 = u^{r_z + r}$. To generate an indistinguishable signature $\tilde{S}$, the simulator randomly chooses $\bar{r}, \tilde{r}, \tilde{y}, \tilde{c}, \tilde{d}$, and then computes $\tilde{g} = g^{\tilde{r}}, \tilde{Z}_0 = S_g^{\tilde{r}}, \tilde{Z}_1 = \tilde{g}^{\tilde{y}}, \tilde{Z}_2 = \tilde{g}^{\tilde{c}}, \tilde{A}_1 = u^{\bar{r}}, \tilde{A}_2 = t^{\tilde{r}}, \tilde{S}_2 = (\tilde{Z}_1 \tilde{Z}_2)^{\tilde{d}}$. Obviously, $\tilde{g}, \tilde{Z}_0, \tilde{A}_1$, and $\tilde{A}_2$ are indistinguishable from $\hat{g}, Z_0, A_1$, and A_2, respectively. Because the EDL problem is intractable, $\tilde{Z}_1, \tilde{Z}_2$ and $\tilde{S}_2$ are indistinguishable from Z_1, Z_2, and S_2, respectively. In addition, S_0 and S_1 are simulatable in the random oracle model. Consequently, the whole signature is simulatable. Hence, the following theorem holds.

Theorem 6.2. *Our signature scheme is signature-simulatable. Thus the properties of unlinkability, anonymity, and non-transferability hold.*

Coalition-resistance. We next show that a colluding subset of group members cannot generate a valid signature that cannot be traced by the group manager. A valid signature S must contain correct S_0, S_1, and S_2. To generate S_2, the colluding members must obtain $S_2 = \xi^d$, where $\xi = \xi_1 \xi_2$ with $\xi_1 = \hat{g}^y$ and $\xi_2 Z_0 = \hat{g}^{v^{e_R}}$. Note that the colluding members cannot derive d even using their signing keys. In addition, the two values ξ_1 and ξ_2 must be computed such that S_0 and S_1 are correct. Here S_0 =SKREP$[(\alpha, \beta) : \hat{g} = g^\beta \wedge Z_0 = S_g^\beta \wedge \xi_1 = \hat{g}^\alpha \wedge A_1 = g^\alpha u^\beta \wedge A_2 = t^\beta](m)$ and S_1 =SKRDL$[\gamma : \xi_2 Z_0 = \hat{g}^{\gamma^{e_R}}](m)$. We now show that the colluding members cannot simultaneously obtain correct S_0, S_1, and S_2. We know that the colluding members cannot compute S_0 and S_1 without knowing α and γ, respectively. Now, to generate an untraceable signature with correct S_0, S_1, and S_2, the colluding members must choose y and c such that $(\hat{g}^{y+c})^d$ and $v = (c+b)^{d_R}$ can be computed. Note that $\xi_1 = \hat{g}^y$, $\xi_2 = \hat{g}^c$, and $\xi = \xi_1 \xi_2 = \hat{g}^{y+c}$. However, the colluding members have no ability to obtain such a c by the following argument. Suppose a group member i has the signing key $(y_i, x_i = g^{c_i}, v_i = (c_i + b)^{d_R}, w_i)$. Because the colluding members cannot compute any c_i, solving for b is infeasible. Thus c' cannot be derived from $(c' + b)$, where $(c' + b)$ is any value that ensures $(c' + b)^{d_R}$ can be computed by the colluding members. As a result, the colluding members cannot compute $(\hat{g}^{y+c})^d$ and $v = (c+b)^{d_R}$ simultaneously. Hence, the following theorem holds.

Theorem 6.3. *Our signature scheme is coalition-resistant.*

7 Conclusions

In this paper, we employ signatures of knowledge and well-known undeniable signature techniques to construct a group undeniable signature scheme. Under reasonable number-theoretic complexity assumptions and the random oracle model, the group undeniable signature scheme is proven to be unforgeable, unlinkable, anonymous, non-transferable, and exculpable. The signature confirmation and denial protocols are zero-knowledge. Even a colluding subset of group members cannot generate valid signatures that cannot be traced.

References:

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

[2] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups (extended abstract). In *Advances in Cryptology—CRYPTO '97*, pp. 410–424, 1997.

[3] D. Chaum. Zero-knowledge undeniable signatures (extended abstract). In *Advances in Cryptology—EUROCRYPT 90*, pp. 458–464, 1990.

[4] D. Chaum and H. van Antwerpen. Undeniable signatures. In *Advances in Cryptology—CRYPTO '89*, pp. 212–216, 1989.

[5] D. Chaum and E. van Heyst. Group signatures. In *Advances in Cryptology—EUROCRYPT 91*, pp. 257–265, 1991.

[6] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, Vol. 13, No. 3, 2000, pp. 361–396.