Group Undeniable Signatures

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

MING-LUEN WU
Department 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 or her identity and the verification of a signature can only be done with the cooperation of the group manager. For business applications, group undeniable signatures can be used to validate price lists, press release or digital contracts when the signatures are commercially sensitive or valuable to a competitor. If a group is falsely accused of having signed a particular signature, the manager should have the ability to prove his innocence. In case of a later dispute, the 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 one message really comes from the alleged sender or not. Generally, a signer keeps a secret value to generate his signature, while opens the corresponding public information for verification. Like human signatures, standard digital signatures must be *nonrepudiatable* and *universal verifiable*. Hence, digital signatures can be extensively applied to digital message as handwritten signatures to paper documents.

However, universal verifiability might not suit the circumstances under which verifying signature is a valuable action. For example, a competitor may inquire about prices and request the merchant to sign the price list. If anyone can verify the signature, the merchant's trade secret will be compromised. Limiting the ability to verify signatures is hence desirable. Chaum and van Antwerpen [4] initiate an *undeniable signature scheme* in which anyone must interact with the signer to verify a valid signature and the signer can disavow an invalid signature through a denial protocol. The important property of non-repudiation still holds because the signer cannot deny his signature except that the signature is indeed invalid.

A *group signature scheme* allows a group member to sign messages on behalf of the group without revealing his or her identity. Nevertheless, in case of a later dispute, a designated group manager can *open* the signature, thus tracing the signer. At the same time, any one—including the group manager—cannot misattribute a valid signature. The concept of group signature schemes is first introduced by Chaum and van Heyst [5], while 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 both nonrepudiatable and universal verifiable.

In this paper, we introduce a new concept, *group undeniable signature scheme*, that is like ordinary group signature schemes except that verifying signatures needs the help of the group manager. The notion of group undeniable signatures combines those of group signatures and undeniable signatures. Applications of group undeniable signatures include validating price lists, press release, or digital contracts when the signatures are commercially sensitive or valuable to a competitor. Our scheme is based on signatures of knowledge [2] and undeniable signature schemes [3]. The proposed scheme can be shown to be existentially unforgeable against adaptive chosen message attacks and be both 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.

This paper is organized as follows. In Section 2, the group undeniable signature model is introduced. Then, in Section 3, useful facts and assumptions in number theory are presented. Section 4 defines basic signatures of knowledge. Section 5 describes our scheme and discusses its security. Conclusions are given in Section 6.

2 Model

In this section we give the definition of a group undeniable signature scheme, the related security requirements, and the significant efficiency considerations. First, we define group undeniable signature schemes. A group undeniable signature scheme consists of the following six components:

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

Join: To become a group member, a user generates his secret key and *membership key*, and registers the membership key with the group manager. Then, the group manager sends to him the *membership certificate*.

Sign: A group member can sign 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 invalid through a signature denial protocol.

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

In general, a group undeniable signature scheme has the following security considerations:

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

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

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

Non-transferability: No one can prove the validity or invalidity of signatures except the group manager.

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 interest parameters:

- 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

We present some number-theoretic results and assumptions. See [7] for additional information.

Notations. For 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 $\gcd(m, n) = 1$. Let $r \in_R I$ represent that r is chosen randomly from a 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 $\mathrm{ord}(g)$.

Fact 3.1. *Let $G =< g >$ 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 $G =< g >$ be a cyclic group generated by g with order n. Next, we present some number-theoretic problems. These problems are assumed to be intractable whether n is known or not.

Discrete Logarithm (DL): Given $y \in_R G$ and the base g, find the discrete logarithm x of $y = g^x$ to the base g.

Representation (Rep): Given $y \in_R G$ and the base g_i for $i = 1, \ldots, k$, find the representation $(x_1, x_2, \ldots, x_k)$ of $y = g_1^{x_1} g_2^{x_2} \cdots g_k^{x_k}$ to the bases $g_1, \ldots, g_k$.

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

Root of Discrete Logarithm (RDL): Given $y \in_R G$, an exponent e and the base g, find the e-th root x of $y = g^{(x^e)}$ to the base g.

The above intractable problems are used for signatures of knowledge described in the next section. Security of our signature scheme is also based on them.

4 Signatures of Knowledge

Signatures of knowledge allow a prover to prove the knowledge of a secret with respect to some public information noninteractively. This cryptographic tool has been used in many group signature schemes. In this section, we review the important signatures of knowledge, which are employed as building blocks in our signature scheme. Now, we explain the notation used in the following signatures of knowledge. Let G be a cyclic group generated by g with order n, where n is an RSA modulus. We denote by Greek letters the elements whose knowledge is proven and by all other letters the elements that are publicly known. Denote by $\|$ the concatenation of two binary strings and by $\wedge$ the conjunction symbol. Assume $\mathcal{H}$ is a collision resistant hash function which maps a binary string of arbitrary length to a hash value of fixed length.

Knowledge of a discrete logarithm. A signature of knowledge of the discrete logarithm of $y = g^x \in G$ to the base g on the message m is a pair (c, s), which can be generated as follows. Choose $r \in_R \mathbb{Z}_n$. Compute

$$c = \mathcal{H}(m \parallel y \parallel g \parallel g^r),$$
$$s = r - cx \bmod n.$$

Such a signature can be computed by a signer who knows the secret x. We denote the signature by $\mathrm{SKDL}[\alpha : y = g^\alpha](m)$. Any one can verify (c, s) by testing $c \stackrel{?}{=} \mathcal{H}(m \parallel y \parallel g \parallel g^s y^c)$.

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 $b_{ij} \in \{1, \ldots, v\}$. A signature of knowledge of a representation $(x_1, \ldots, x_u)$ of $y_1, \ldots, y_w$ with respect to the bases $g_1, \ldots, g_v$ on the message m is $(c, s_1, s_2, \ldots, s_u)$, which can be generated as follows. Choose $r_i \in_R \mathbb{Z}_n$ for $i = 1, \ldots, u$. Compute

$$\begin{aligned} c =& \mathcal{H}(m \parallel y_1 \parallel \ldots \parallel y_w \parallel g_1 \parallel \ldots \parallel g_v \\ & \parallel \{\{e_{ij}, b_{ij}\}_{j=1}^{\ell_i}\}_{i=1}^{w} \parallel \prod_{j=1}^{\ell_1} g_{b_{1j}}^{r_{e_{1j}}} \parallel \cdots \parallel \prod_{j=1}^{\ell_w} g_{b_{wj}}^{r_{e_{wj}}}), \\ s_i =& r_i - cx_i \bmod n, \text{for } i = 1, \ldots, u. \end{aligned}$$

Such a signature can be computed by a signer who knows a representation $(x_1, \ldots, x_u)$. We denote this signature by

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

Any one can verify the signature by testing $c \stackrel{?}{=} \mathcal{H}(m \parallel y_1 \parallel \ldots \parallel g_v \parallel \{\{e_{ij}, b_{ij}\}_{j=1}^{\ell_i}\}_{i=1}^{w} \parallel \prod_{j=1}^{\ell_1} g_{b_{1j}}^{s_{e_{1j}}} y_1^c \parallel \cdots \parallel \prod_{j=1}^{\ell_w} g_{b_{wj}}^{s_{e_{wj}}} y_w^c)$.

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 $v = f^w g^{x^e} \in G$ to the bases f and g. A signature of knowledge of the pair (w, x) of $v = f^w g^{x^e}$ on the message m consists of two components:

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

To generate the signature efficiently, a small integer e is chosen. 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, according to the equations $v_i = f^{r_i} g^{x^i}$ and $v = f^w g^{x^e}$, we actually have $\gamma_1 = r_1, \gamma_i = r_i - x\gamma_{i-1}$ for $i = 2, \ldots, e-1$, $\gamma_e = w - x\gamma_{e-1}$, and $\delta = x$. Hence, the second component can be obtained. We denote this whole signature by

$$\mathrm{SKRREP}[(\alpha, \beta) : v = f^\alpha g^{\beta^e}](m).$$

Knowledge of roots of discrete logarithms. Let e be a small integer. Assume f is also a generator of G and $\log_g f$ is not known. A signature of knowledge of the e-th root x of the discrete logarithm of $y = g^{x^e}$ to the base g on the message m comprises two components:

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

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

According to the further results in [6, Section 3], in the random oracle model, the signatures SKDL and SKREP are simulatable and they are existentially unforgeable against adaptive chosen message attacks under the related number-theoretic complexity assumptions. Thus, SKRREP and SKRDL clearly have the same properties.

5 The Scheme

Now we propose our scheme and discuss its security.

5.1 System Setup

To derive the group secret and group public keys, 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 $G =< g >$ of order n,

- $f = g^a, S_f = f^d, S_g = g^b, u = g^h, t = u^\rho$ where a, d, b, h, and $\rho \in_R \mathbb{Z}_n^*$,
- (e, d) for $e, d \in_R \mathbb{Z}_n^*$ such that $ed \equiv 1 \pmod{n}$.

It is noteworthy that n must be chosen such that factoring n and solving DL in G are intractable. Assume g_0 is a generator of $\mathbb{Z}_p^*$, where p is a prime. G could be a subgroup of $\mathbb{Z}_p^*$ with generator $g_0^{(p-1)/n}$, where $n \mid (p-1)$. By Fact 3.1, $g = g_0^{(p-1)/n}$ has order n. Moreover, the order of f, S_f, S_g, u, and t is 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 one, say Alice, wants to join the group, she chooses the secret key $y \in_R \mathbb{Z}_n^*$ and computes 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 y. Next, the group manager chooses $c \in_R \mathbb{Z}_n^*$ such that $\gcd(y + c, n) = 1$, computes Alice's membership certificate $(x = g^c, v = (c + b)^{d_R} \bmod n, w = (zx)^d)$, and sends (x, v, w) to Alice. Such a (y, x, v, w) is called a *valid signing key*. It is important to note that the group manager must choose distinct c's for different registers and prevent anyone from knowing c's. In addition , by Fact 3.1, we have $\mathrm{ord}(z) = \mathrm{ord}(x) = \mathrm{ord}(w) = n$.

5.3 Sign

Given a message m, Alice can generate the signature S by computing the following nine values:

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

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 the conjunction of the correctness of S_0, S_1, and S_2.

Now we explain the roles of the elements in S. First, considering S_0, it proves that the same random number is used in the computation of $\hat{g}, Z_0, A_1$, and A_2, and proves that the same exponent y' is used in $Z_1 = \hat{g}^{y'}$ and $A_1 = g^{y'} u^r$ for some $y' \in_R \mathbb{Z}_n^*$. If S_0 is correct, (A_1, A_2) is an ElGamal encryption of $z = g^{y'}$ with respect to the group public key (u, t). The element S_1 proves that Alice knows the knowledge of an e_R-th root of the discrete logarithm of $Z_2 Z_0$ to the base $\hat{g}$. Finally, considering S_2, the verifier must interact with the group manager to check whether $S_2 = (Z_1 Z_2)^d$ or not.

5.4 Signature Confirmation Protocol

A signature confirmation protocol is an interactive protocol between the group manager and a verifier, in which the group manager can convince a verifier of the fact that a 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 sequel, we denote by $\mathcal{P}$ the group manager and by $\mathcal{V}$ the verifier. Let $X \longrightarrow Y : Z$ represent 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. Now, we present how $\mathcal{V}$ can 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 $\mathcal{V}$ do 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 $(Z_1 Z_2)^{e_1} f^{e_2} \stackrel{?}{=} B$.
   If equality holds then $\mathcal{V}$ accepts S as a valid signature for m. Otherwise S is undetermined.

Our confirmation protocol is based on Chaum's method [3]. 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.

In the following theorem, we prove that the verifier will accept a valid signature.

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 prove that the group manager cannot cheat a verifier into accepting an invalid signature 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, $S_2 \neq (Z_1Z_2)^d$. $\mathcal{P}$ can make $\mathcal{V}$ accept the signature only if $\mathcal{P}$ can find $B = (Z_1Z_2)^{e_1}f^{e_2}$ 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:

$$A = S_2^{e_1}S_f^{e_2} \quad (1)$$

$$B = (Z_1Z_2)^{e_1}f^{e_2}, \quad (2)$$

where $S_2 \neq (Z_1Z_2)^d$. Assume $A = f^i, B = f^j, S_2 = f^k$, and $Z_1Z_2 = f^\ell$ for $i, j, k, \ell \in Z_n$. Recall $S_f = f^d$. From (1) and (2), we have

$$i = ke_1 + de_2 \bmod n \quad (3)$$

$$j = \ell e_1 + e_2 \bmod n. \quad (4)$$

Because $f^k \neq f^{\ell d}, k \neq \ell d \pmod n$. As a result, there is only one solution for (e_1, e_2) from (3) and (4).
By Fact 3.1, the order of S_2, S_f, and Z_1Z_2 is n. Hence, there are n ordered pairs (e_1, e_2) corresponding to A. $\mathcal{P}$ can not identify which of them has been used to compute A by $\mathcal{V}$. In addition, every B is the correct response for exactly one of the possible ordered pairs. Consequently, the probability that $\mathcal{P}$ will give $\mathcal{V}$ the correct response B verified is $1/n$. The theorem is proven. □

5.5 Signature Denial Protocol

A signature denial protocol is an interactive protocol between $\mathcal{P}$ and $\mathcal{V}$, which allows $\mathcal{P}$ to convince $\mathcal{V}$ of the fact that an alleged signature is 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, 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. Now, we present how $\mathcal{P}$ can make $\mathcal{V}$ accept an invalid signature S as invalid. 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 $\mathcal{V}$ repeat the following steps at most c_2 times. When $\mathcal{V}$ finds S is undetermined, the protocol stops.

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. $\mathcal{P} \longrightarrow \mathcal{V} : B$
   $\mathcal{P}$ computes $A_1/A_2^e = (Z_1Z_2/S_2^e)^{e_1}$. $\mathcal{P}$ finds e_1, and then sends $B = e_1$ to $\mathcal{V}$.
3. $\mathcal{V}$ checks whether $B \stackrel{?}{=} e_1$.
   If equality holds, then $\mathcal{V}$ is convinced that S is invalid one time. Otherwise S is undetermined.

If convinced of S's invalidity c_2 times, $\mathcal{V}$ will accept S as invalid. It is noteworthy that $\mathcal{P}$ can perform at most c_1c_2 operations to find the correct e_1's.

The denial protocol is based on Chaum's method [3]. 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.

In the following theorem, we prove $\mathcal{P}$ can convince $\mathcal{V}$ of the fact that an alleged signature is 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 correct. Because S is generated without a valid signing key, $S_2 \neq (Z_1Z_2)^d$. Therefore $S_2^e \neq Z_1Z_2$. We have $A_1/A_2^e = (Z_1Z_2/S_2^e)^{e_1}$. Consequently, $\mathcal{P}$ can always find e_1 and give the correct response. This implies that $\mathcal{V}$ will accept S as an invalid signature for m. □

Next, we prove that $\mathcal{P}$ cannot fool $\mathcal{V}$ into accepting a valid signature as invalid 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 $1/c_1^{c_2}$. □

5.6 Open

Given a valid signature S, the group manager can compute $z_P = A_1A_2^{-\rho^{-1}}$. The signer with the membership key z_P 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}.

5.7 Security Analysis

The security notions below are considered under reasonable number-theoretic complexity assumptions and the random oracle model.

Exculpability. Because the DL problem is intractable, neither the group manager nor a group member can compute the secret key of another group member. Thus, it is infeasible to frame another member. Note this does not prevent the group manager from generating any valid signatures.

Unforgeability. We prove that our signature is existentially unforgeable against adaptive chosen message attacks. Recall that any valid signature $\bar{S}$ 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 $\xi_1 = \bar{g}^{\bar{y}}$, $\xi_2\bar{Z}_0 = \bar{g}^{\bar{v}^{e_R}}$. 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$[(\alpha, \beta) : \bar{g} = g^\beta \wedge \bar{Z}_0 = S_g^\beta \wedge \xi_1 = \bar{g}^\alpha \wedge \bar{A}_1 = g^\alpha u^\beta \wedge \bar{A}_2 = t^\beta](m)$ and S_1 =SKRDL$[\gamma : \xi_2 \bar{Z}_0 = \bar{g}^{\gamma^{e_R}}](m)$. Next, we show that the attacker cannot simultaneously obtain correct S_0, S_1 and S_2. Suppose $\alpha = \bar{y}$ and $\gamma = \bar{v}$. Note that the attacker cannot compute S_0 and S_1 without knowing $\bar{y}$ and $\bar{v}$, respectively. Now, to obtain S_0 from a (ξ, ξ^d), the attacker chooses $\bar{y}$ and has $\xi_1 = \bar{g}^{\bar{y}}$. So $\xi_2 = \xi\xi_1^{-1}$. Assume $\xi_2 = \bar{g}^{\bar{c}}$. Because the value $\bar{v} = (\bar{c} + b)^{d_R} \bmod n$ satisfying $\xi_2 \bar{Z}_0 = \bar{g}^{\bar{v}^{e_R}}$ cannot be obtained, S_1 is existentially unforgeable against adaptive chosen message attacks. Consequently, we have the following theorem:

Theorem 5.5. *Our signature scheme is existentially unforgeable against adaptive chosen message attacks.*

Unlinkability, Anonymity, Non-traceability. These properties hold if the signatures are simulatable. Now, we show the signatures can be simulated. 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. Recall that S_0 and S_1 are simulatable in the random oracle model. Consequently, the whole signature is simulatable. Hence, we have the following theorem:

Theorem 5.6. *Our signature scheme is signature-simulatable. Thus the properties of unlinkability, anonymity, and non-traceability hold.*

Coalition-resistance. We show that a colluding subset of group members cannot generate a valid signature that cannot be traced by the group manager. A valid signature $\bar{S}$ must contain correct S_0, S_1, and S_2. Considering S_2, the colluding members must obtain $S_2 = \xi^d$, where $\xi = \xi_1\xi_2$ with $\xi_1 = \bar{g}^{\bar{y}}$, $\xi_2 \bar{Z}_0 = \bar{g}^{\bar{v}^{e_R}}$. However, even using their signing keys, the colluding members cannot derive d; they can obtain $\xi = g^r$ and ξ^d for any r. 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$[(\alpha, \beta) : \bar{g} = g^\beta \wedge \bar{Z}_0 = S_g^\beta \wedge \xi_1 = \bar{g}^\alpha \wedge \bar{A}_1 = g^\alpha u^\beta \wedge \bar{A}_2 = t^\beta](m)$ and S_1 =SKRDL$[\gamma : \xi_2 \bar{Z}_0 = \bar{g}^{\gamma^{e_R}}](m)$. In the following, we show that the colluding members cannot simultaneously obtain correct S_0, S_1, and S_2. Suppose $\alpha = \bar{y}$ and $\gamma = \bar{v}$. We know that the colluding members cannot compute S_0 and S_1 without knowing $\bar{y}$ and $\bar{v}$, respectively. Now, to generate an untraceable signature with correct S_0, S_1, and S_2, the colluding members must choose $\bar{y}$ and $\bar{c}$ such that $(\bar{g}^{\bar{y}+\bar{c}})^d$ and $\bar{v} = (\bar{c} + b)^{d_R}$ can be computed. Note that $\xi_1 = \bar{g}^{\bar{y}}$, $\xi_2 = \bar{g}^{\bar{c}}$, and $\xi = \xi_1\xi_2 = \bar{g}^{\bar{y}+\bar{c}}$. However, we will show that the colluding members have no ability to obtain such a $\bar{c}$. 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 $\bar{c}'$ cannot be derived from $(\bar{c}' + b)$, where $(\bar{c}' + b)$ is any value that ensures $(\bar{c}' + b)^{d_R}$ can be computed by the colluding members. As a result, the colluding members cannot compute $(\bar{g}^{\bar{y}+\bar{c}})^d$ and $\bar{v} = (\bar{c}+b)^{d_R}$ simultaneously. Hence, we have the following theorem:

Theorem 5.7. *Our signature scheme is coalition-resistant.*

6 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, we can prove the group undeniable signature scheme is 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*, pages 62–73, 1993.

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

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

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

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

[6] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

[7] K. H. Rosen. *Elementary Number Theory and its Applications (Third Edition)*. Addison Wesley, 1993.