Convertible Group Undeniable Signatures

Yuh-Dauh Lyuu[1] and Ming-Luen Wu[2]

[1] Dept. of Computer Science & Information Engineering and Dept. of Finance,
National Taiwan University, Taiwan
lyuu@csie.ntu.edu.tw
[2] Dept. of Computer Science & Information Engineering,
National Taiwan University, Taiwan
d5526009@csie.ntu.edu.tw

Abstract. Group undeniable signatures are like ordinary group signatures except that verifying signatures needs the help of the group manager. In this paper, we propose a convertible group undeniable signature scheme in which the group manager can turn all or selective signatures, which are originally group undeniable signatures, into ordinary group signatures without compromising security of the secret key needed to generate signatures. The proposed scheme also allows the group manager to delegate the ability to confirm and deny to a limited set of parties without providing them the capability of generating signatures. For business applications, convertible group undeniable signatures can be widely used to validate price lists, press release or digital contracts when the signatures are commercially sensitive or valuable to a competitor. Our scheme is 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.

1 Introduction

In electronic life, digital signatures are used to verify whether one message really comes from the alleged signer or not. Like human signatures, standard digital signatures must be *nonrepudiatable* and *universally verifiable*. However, universal verifiability might not suit the circumstances under which verifying signature is a valuable action. Chaum and van Antwerpen [5] 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 disavow the signature through a denial protocol except that the signature is indeed invalid.

With undeniable signatures, anyone needs the cooperation of the signer for verifying the signatures. This is not satisfactory because the signer might pass away or be occupied. Boyar et al. [2] first introduce the concept of *convertible undeniable signatures*: By releasing appropriate verification keys, the signer can turn all or selective signatures, which are original undeniable signatures, into ordinary digital signatures without compromising security of the secret key needed to generate signatures. The convertible schemes in [2, 7] consider converting valid undeniable signatures to universal verifiable ones. Michels and Stadler [9] present a convertible undeniable signature scheme in which the signer can not only convert valid undeniable signatures into ordinary signatures, but also convert invalid undeniable signatures into universal verifiable statements about the fact.

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 initiated by Chaum and van Heyst [6], while Camenisch and Stadler [3] 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.

A *group undeniable signature* is like an ordinary group signature except that verifying signatures needs the help of the group manager. In this paper, we propose a *convertible group undeniable signature scheme* in which the group manager can convert all or selective signatures, which are originally group undeniable signatures, into universally verifiable ones without compromise of security of the signing key. The notions of convertible group undeniable signatures combine those of group signatures [6] and convertible undeniable signatures [9]. The proposed scheme also allows the group manager to delegate the ability to confirm and deny to a limited set of parties without providing them the capability of generating signatures. Our scheme is based on signatures of knowledge [3] and undeniable signature schemes [4]. We can show the present scheme is existentially unforgeable against adaptive chosen message attacks and is both signature-simulatable and coalition-resistant under reasonable number-theoretical 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 convertible 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 convertible 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.

A convertible group undeniable signature scheme is a group undeniable signature scheme with the following additional components:

Individual receipt generation: Given a message, an alleged signature and the group secret key, the group manager can generate the individual receipt by which anyone can verify whether the alleged signature is valid or not. A group undeniable signature can be converted into an ordinary group signature by releasing its individual receipt.

Individual verification. Given a message, an alleged signature, an individual receipt, and the group public key, one can check the receipt is valid or invalid with respect to the alleged signature; in case of the former, the alleged signature can be verified using the receipt.

Universal receipt generation. With the group secret key, the group manager can generate the universal receipt by which anyone can verify whether signatures are valid or not. A group undeniable signature scheme can totally be converted into an ordinary group signature scheme by releasing the universal receipt.

Universal verification. With the group public key, one can check the given universal receipt is valid or invalid. Suppose the receipt is valid. Given a message and an alleged signature, anyone can verify the signature using the receipt.

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 [11, 12] 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 $m \in \{1, 2, \dots, n-1\}$ 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 $g^d \equiv 1 \pmod{n}$ is called the *order* of g modulo n, and is denoted by $\text{ord}_n g$ or $\text{ord}(g)$. A *universal exponent* of n is a positive integer u such that $g^u \equiv 1 \pmod{n}$ for all g relatively prime to n. The minimal universal exponent of n is denoted by $\lambda(n)$.

Fact 1. *If q and $p = 2q + 1$ are both primes and a is a positive integer with $1 < a < p - 1$, then $-a^2$ is a quadratic nonresidue and a primitive root modulo p.*

Fact 2. *Let M be a positive integer with odd prime factorization $M = p_1 p_2 \cdots p_n$.* (1)

$$\lambda(M) = \text{lcm}(\phi(p_1), \phi(p_2), \dots, \phi(p_n)).$$

(2) *There exists an integer g such that* $\text{ord}_M g = \lambda(M)$*, the largest possible order of an integer modulo M.* (3) *Let r_i be a primitive root modulo p_i. The solution of simultaneous congruences $x \equiv r_i \pmod{p_i}$, $i = 1, 2, \dots, n$, produces such an integer g.*

Fact 3. *Let $G =< g >$ be a cyclic group generated by g. If $ord(g) = n$ and if r is a positive integer, then*

$$ord(g^r) = n/\gcd(n, r)$$

Thus, if we choose a positive integer a such that $\gcd(a, n) = 1$, then g^a has the same order as g.

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 Logarithm (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 M, where M 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 $r \in \mathbb{Z}$. Compute

$$\begin{aligned} c &= \mathcal{H}(m \parallel y \parallel g \parallel g^r), \\ s &= r - cx. \end{aligned}$$

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 $c \overset{?}{=} \mathcal{H}(m \parallel y \parallel g \parallel g^s y^c)$.

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 $r_i \in \mathbb{Z}$ for $i = 1, \ldots, u$. Compute

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

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

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

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

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 $r_i \in \mathbb{Z}$,
- $\text{SKREP}[(\gamma_1, \gamma_2 \ldots, \gamma_e, \delta) : v_1 = f^{\gamma_1} g^{\delta} \wedge v_2 = f^{\gamma_2} v_1^{\delta} \wedge \cdots \wedge v_{e-1} = f^{\gamma_{e-1}} v_{e-2}^{\delta} \wedge v = f^{\gamma_e} v_{e-1}^{\delta}](m)$.

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 $r_i \in_R \mathbb{Z}$, 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

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

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:

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

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 [10, 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 present 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:

- $n = p_1p_2$, where both $p_i = 2q_i + 1$ and q_i are primes for $i = 1, 2$,
- an RSA public key (q_1q_2, e_R) and secret key d_R,
- an integer $g \in \mathbb{Z}_n^*$ such that $\mathrm{ord}_n g = q_1q_2$,
- $f = g^a, S_f = f^d, S_g = g^b, u = g^h, t = u^\rho$, where $a, d, b, h, \rho \in_R \mathbb{Z}_{q_1q_2}^*$, and all arithmetic is modulo n,
- (e, d) for $e, d \in_R \mathbb{Z}_{q_1q_2}^*$ such that $ed \equiv 1 \pmod{q_1q_2}$,

It is noteworthy that n must be chosen such that factoring n and solving DL in $\mathbb{Z}_n^*$ are intractable. By Fact 1 and 2, we can obtain g_0 with order $\lambda(n) = 2q_1q_2$, and then have $g = g_0^2$ with order q_1q_2 by Fact 3. Moreover, the order of f, S_f, S_g, u, and t is also q_1q_2. 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 $z = g^y \bmod n$. We can assume that $\gcd(y, q_1q_2) = 1$. 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 $c \in \mathbb{Z}_{q_1q_2}^*$ such that $(zg^c)^{q_1} \neq 1 \pmod{n}$ and $(zg^c)^{q_2} \neq 1 \pmod{n}$ (this can be done by testing at most three continuous integers). Note that $\gcd(y + c, q_1q_2) = 1$. Then the group manager computes Alice's membership certificate $(x = g^c \bmod n, v = (c + b)^{d_R} \bmod q_1q_2, w = (zx)^d \bmod n)$, 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, we have $\mathrm{ord}(z) = \mathrm{ord}(x) = \mathrm{ord}(w) = q_1q_2$.

5.3 Sign

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

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

The above arithmetic is modulo n. 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 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 $A = S_2^{e_1} S_f^{e_2} \bmod n$.
2. $\mathcal{P} \longrightarrow \mathcal{V} : B$
   $\mathcal{P}$ computes $B = A^e \bmod n$.

3. $\mathcal{V}$ verifies that $(Z_1Z_2)^{e_1} f^{e_2} \overset{?}{=} B \bmod n$.
   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 [4]. 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 $B = (Z_1Z_2)^{e_1} f^{e_2} \bmod n$ 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.

We prove that the verifier will accept a valid signature.

Theorem 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 $w = (g^{y+c})^d \bmod n$, we have

$$S_2 \equiv w^r \equiv ((g^{y+c})^d)^r \equiv ((\hat{g})^{y+c})^d \equiv (Z_1Z_2)^d \pmod{n}.$$

So $B \equiv A^e \equiv ((S_2)^{e_1}(S_f)^{e_2})^e \equiv (Z_1Z_2)^{e_1} f^{e_2} \pmod{n}$. □

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 2. *If S is not a valid group undeniable signature, then a verifier will accept S as a valid signature for m with probability at most $1/q_1q_2$.*

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, $S_2 \neq (Z_1Z_2)^d \bmod n$. $\mathcal{P}$ can make $\mathcal{V}$ accept the signature only if $\mathcal{P}$ can find $B = (Z_1Z_2)^{e_1} f^{e_2} \bmod n$ such that (e_1, e_2) satisfies $A \equiv S_2^{e_1}(S_f)^{e_2} \pmod{n}$. That is, (e_1, e_2) satisfies the following two equations:

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

$$B = (Z_1Z_2)^{e_1} f^{e_2} \bmod n, \tag{2}$$

where $S_2 \neq (Z_1Z_2)^d \bmod n$. Assume $A = f^i, B = f^j, S_2 = f^k$, and $Z_1Z_2 = f^\ell$, where $i, j, k, \ell \in Z_{q_1q_2}$, and all arithmetic is modulo n. Recall $S_f = f^d \bmod n$. From (1) and (2), we have

$$i = ke_1 + de_2 \bmod q_1q_2 \tag{3}$$

$$j = \ell e_1 + e_2 \bmod q_1q_2. \tag{4}$$

Because $f^k \neq f^{\ell d} \pmod{n}, k \neq \ell d \pmod{q_1q_2}$. As a result, there is only one solution for (e_1, e_2) from (3) and (4).

By Fact 3, the order of S_2, S_f, and Z_1Z_2 is q_1q_2. Hence, there are at least q_1q_2

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 $q_1 q_2$ ordered pairs (e_1, e_2) for $e_1, e_2 < q_1 q_2$. Consequently, the probability that $\mathcal{P}$ will give $\mathcal{V}$ the correct response B verified is at most $1/q_1 q_2$. 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 $A_1 = (Z_1 Z_2)^{e_1} f^{e_2} \bmod n$, $A_2 = S_2^{e_1} S_f^{e_2} \bmod n$.
2. $\mathcal{P} \longrightarrow \mathcal{V} : B$
   $\mathcal{P}$ computes $A_1/A_2^e \equiv (Z_1 Z_2/S_2^e)^{e_1} \pmod n$. $\mathcal{P}$ finds e_1, and then sends $B = e_1$ to $\mathcal{V}$.
3. $\mathcal{V}$ checks whether $B \overset{?}{=} e_1$.
   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 $c_1 c_2$ operations to find the correct e_1's.

The denial protocol is based on Chaum's method [4]. 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 $A_1 = (Z_1 Z_2)^{e_1} f^{e_2} \bmod n$ and $A_2 = S_2^{e_1} S_f^{e_2} \bmod n$, 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 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, $S_2 \neq (Z_1 Z_2)^d \bmod n$. Therefore $S_2^e \neq Z_1 Z_2$. We have $A_1/A_2^e \equiv (Z_1 Z_2/S_2^e)^{e_1} \pmod{n}$. 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 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 $S_2 = (Z_1 Z_2)^d \bmod n$. Therefore $S_2^e \equiv Z_1 Z_2 \pmod{n}$. We have $A_1/A_2^e \equiv (Z_1 Z_2/S_2^e)^{e_1} \equiv 1 \pmod{n}$. 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 $z_P = A_1 A_2^{-\rho^{-1}}$. The signer with the membership key $z = z_P$ 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 Convertibility

We describe the four components for convertibility.

Individual receipt generation. Let S be a signature for the message m. We show how to generate its individual receipt. The group manager chooses $r \in_R \mathbb{Z}_{q_1 q_2}^*$, and computes the receipt $R = (\tilde{f}, R_1, R_2, R_3)$ as follows:

$$\begin{aligned}
\tilde{f} &= f^r \bmod n,\\
R_1 &= (Z_1 Z_2)^r \bmod n,\\
H &= \mathcal{H}(m \parallel \tilde{f} \parallel R_1),\\
R_2 &= \mathrm{SKREP}[\alpha : R_1 = (Z_1 Z_2)^\alpha \wedge \tilde{f} = f^\alpha](m),\\
R_3 &= r - Hd \bmod q_1 q_2.
\end{aligned}$$

Obviously, releasing the individual receipt does not compromise security of the secret key d needed to generate signatures.

Individual verification. To check R, one sees the correctness of R_2 and tests whether $\tilde{f} = f^{R_3} S_f^H \bmod n$. If both succeed, then the receipt R with respect to S is valid. Otherwise the receipt is invalid. If R is valid, then the alleged signature S can be verified by checking the correctness of S_0 and S_1, and testing whether $R_1 = (Z_1 Z_2)^{R_3} S_2^H \bmod n$. Hence, with the individual receipt R, the alleged signature S can be universally verified.

Universal receipt generation. To make all signatures universally verifiable, the group manager releases e as the universal receipt. According to the basic assumption behind regular RSA, this does not compromise security of the secret key d needed to generate signatures.

Universal verification. To check e, one can test whether $f = S_f^e \bmod n$. If the equality holds, then e is valid. Otherwise e is invalid. If e is valid, then all alleged signatures can be verified by checking the correctness of S_0 and S_1, and testing whether $Z_1 Z_2 \equiv S_2^e \pmod{n}$. Consequently, the group undeniable signature scheme can totally be converted into an ordinary group signature scheme by releasing the universal receipt e. In addition, our scheme allows the group manager to delegate the ability to confirm and deny to a limited set of parties by issuing e only to them.

5.8 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. However, 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 $S_2 = \xi^d \bmod n$, where $\xi = \xi_1 \xi_2$ with $\xi_1 = \bar{g}^{\bar{y}} \bmod n$, $\xi_2 \bar{Z}_0 = \bar{g}^{\bar{v}^{e_R}} \bmod n$. 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 $\xi_1 = \bar{g}^{\bar{y}} \bmod n$. So $\xi_2 = \xi \xi_1^{-1} \bmod n$. Assume $\xi_2 = \bar{g}^{\bar{c}} \bmod n$. Because the value $\bar{v} = (\bar{c} + b)^{d_R}$ satisfying $\xi_2 \bar{Z}_0 \equiv \bar{g}^{\bar{v}^{e_R}} \bmod n$ cannot be obtained, S_1 is existentially unforgeable against adaptive chosen message attacks. Consequently, we have the following theorem:

Theorem 5. *Our signature scheme is existentially unforgeable against adaptive chosen message attacks.*

Unlinkability, Anonymity, Non-transferability. These properties hold if the signatures are simulatable. Now, we show the signatures can be simulated. Let S be a valid signature. Assume the signer's membership key z equals $u^{r_z} \bmod n$ for some $r_z \in \mathbb{Z}_n^*$. So $A_1 = u^{r_z + r} \bmod n$. To generate an indistinguishable signature $\tilde{S}$, the simulator randomly chooses $\bar{r}, \tilde{r}, \tilde{y}, \tilde{c}, \tilde{d}$, and then computes $\tilde{g} = g^{\tilde{r}}, \tilde{Z}_0 = S_g^{\tilde{r}}, \tilde{Z}_1 = \tilde{g}^{\tilde{y}}, \tilde{Z}_2 = \tilde{g}^{\tilde{c}}, \tilde{A}_1 = u^{\bar{r}}, \tilde{A}_2 = t^{\tilde{r}}, \tilde{S}_2 = (\tilde{Z}_1 \tilde{Z}_2)^{\tilde{d}}$, where all arithmetic is modulo n. 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 6. *Our signature scheme is signature-simulatable. Thus the properties of unlinkability, anonymity, and non-traceability hold.*

Zero knowledge. By applying the commitment techniques, the confirmation and denial protocols reveal no extra information except for the validity or invalidity of a signature. As a result, our scheme can be zero-knowledge.

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, colluding members must obtain $S_2 = \xi^d \bmod n$, where $\xi = \xi_1 \xi_2$ with $\xi_1 = \bar{g}^{\bar{y}} \bmod n$, $\xi_2 \bar{Z}_0 = \bar{g}^{\bar{v}^{e_R}} \bmod n$. However, even using their signing keys, the colluding members cannot derive d; they can obtain $\xi = g^r \bmod n$ and $\xi^d \bmod n$ 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$[\gamma : \xi_2 \bar{Z}_0 = \bar{g}^{\gamma^{e_R}}](m)$. Next, 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 obtain the correct S_0, S_1 and S_2, the colluding members must choose $\bar{y}$ and $\bar{c}$ such that $\bar{y}+\bar{c}$ and $\bar{v} = (\bar{c}+b)^{d_R}$ can be computed. Note that $\xi_1 = \bar{g}^{\bar{y}} \bmod n$, $\xi_2 = \bar{g}^{\bar{c}} \bmod n$, and $\xi \equiv \xi_1 \xi_2 \equiv \bar{g}^{\bar{y}+\bar{c}} \pmod{n}$. In the following we show that obtaining such a $\bar{c}$ is infeasible. Suppose a group member i has the signing key $(y_i, x_i = g^{c_i} \bmod n, v_i = (c_i + b)^{d_R} \bmod q_1 q_2, w_i)$. Because the colluding members cannot compute the c_i's, solving for b is infeasible. Therefore $\bar{c}'$ cannot be derived from $(\bar{c}' + b)$, where $(\bar{c}' + b)$ is any value such that $(\bar{c}' + b)^{d_R}$ can be obtained by the colluding members. As a result, $\bar{y} + \bar{c}'$ cannot be computed. This implies that it is infeasible to choose $\bar{y}$ and $\bar{c}$ such that $\bar{y} + \bar{c}$ and $\bar{v} = (\bar{c} + b)^{d_R}$ are derived simultaneously. Now, we have the following theorem:

Theorem 7. *Our signature scheme is coalition-resistant.*

6 Conclusions

In this paper, we employ signatures of knowledge and RSA-based undeniable signature techniques to construct a convertible group undeniable signature scheme. Our scheme also allows the group manager to delegate the ability to confirm and deny to a limited set of parties without providing them the capability of generating signatures. 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. Boyar, D. Chaum, I. Damgård, and T. Pedersen. Convertible undeniable signatures. In *Advances in Cryptology—CRYPTO '90*, pages 189–205, 1990.
3. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups (extended abstract). In *Advances in Cryptology—CRYPTO '97*, pages 410–424, 1997.
4. D. Chaum. Zero-knowledge undeniable signatures (extended abstract). In *Advances in Cryptology—EUROCRYPT 90*, pages 458–464, 1990.
5. D. Chaum and H. van Antwerpen. Undeniable signatures. In *Advances in Cryptology—CRYPTO '89*, pages 212–216, 1989.
6. D. Chaum and E. van Heyst. Group signatures. In *Advances in Cryptology—EUROCRYPT 91*, pages 257–265, 1991.
7. I. Damgård and T. Pedersen. New convertible undeniable signature schemes. In *Advances in Cryptology—EUROCRYPT 96*, pages 372–386, 1996.
8. S. J. Kim, S. J. Park and D. H. Won. Convertible group signatures. In *Advances in Cryptology—ASIACRYPT 96*, pages 311–321, 1996.
9. M. Michels and M. Stadler. Efficient convertible undeniable signature schemes. In *Proc. 4th Workshop on Selected Areas in Cryptography (SAC '97)*, pages 231–244, 1997.
10. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
11. K. H. Rosen. *Elementary Number Theory and its Applications (Third Edition)*. Addison Wesley, 1993.
12. H. N. Shapiro. *Introduction to the Theory of Numbers.* John Wiley & Sons, 1983.