An Interesting Member ID-based Group Signature

Sujing Zhou, Dongdai Lin
SKLOIS Lab,Institute of Software,
Chinese Academy of Sciences, P.R. China
Email: {zhousujing,ddlin}@is.iscas.ac.cn

Abstract

We propose an interesting efficient member ID-based group signatures, i.e., verification of output from algorithm OPEN run by the group manager does not have to refer to a registration table (acting as certification list).

The proposal is free of GM-frameability, i.e., secret key of member is not escrowed to GM, which is unique among all known member ID-based group signatures as far as we know.

The proposal also has two distinguished extra features, one is that the group manager does not have to maintain a registration table to obtain the real identity of the signer in contrast to other schemes, another is that it provides an alternative countermeasure against tampered registration table to applying integrity techniques to the table in case registration table is maintained.

Keywords: Digital Signature; Group Signature; Identity based; Partial trapdoor one-way function.

1 Introduction

1.1 Group Signature

Group signature schemes [1] are motivated by enabling members of a group to sign on behalf of the group without leaking their true identities; but the signer's identity is able to be opened, i.e., discovered by the group manager (GM for short) on disputes. Group signatures have been found useful in various applications, e.g. anonymous authentication, internet voting, electronic bidding.

A group signature, however, can be viewed as a proof of knowledge of one of the secret key $(sk_1, ..., sk_n)$ corresponding to a list of public keys $(pk_1, ..., pk_n)$ [2], or a proof of knowledge of a signature (also called group member certificate) signed by GM [3, 4, 5, 6, 7, 8, 9, 10], in contrast with an ordinary signature scheme which is the counterpart of handwritten signature in the digital world and can be viewed as a proof of knowledge of secret key *sk* corresponding to the public key *pk*. The counterpart of a group signature in the real world is official seal, at the sight of which, anyone can be assured that it is made by some person from the claimed authority, but have no idea of who that person is.

In further details, a group signature scheme is composed of the following steps:

(1) GM, the group manager, firstly chooses the security parameters as well as a group secret key and a group public key.

(2) Any group member candidate is required to choose his *member secret key*, and run an interactive protocol with GM to join in the group, during which GM generates a signature (also called *member certificate*) on the member secret key blindly, i.e., without knowing the secret key value.

(3) Any group member can generate group signatures using his member secret key and member certificate, called *group signing key* all together. In most efficient group signatures [3, 4, 5, 6, 7, 8, 9, 10], a group signature is generated by applying Fiat-Shamir's heuristic method [11] to a zero-knowledge proof of knowledge of a member secret key and the corresponding member certificate.

(4) A group signature is verified with respect to the group public key according to the prescribed algorithm VERIFY. The identity of the group member who has generated the group signature is unavailable to any one except GM.

(5) On disputes, e.g., some members of the group are suspicious of abusing their authority to issue licences to ineligible persons, GM is able to find out the identity of the group member by "opening" the group signature, i.e., executing an algorithm with group secret key and the group signature as inputs.

(6) The attribution to some group member should be convincing, i.e., the output of the above "opening" process is judgeable according to the prescribed algorithm JUDGE.

Common Enhancements. In some applications, GM is required to be two independent authorities, one issuing member certificates (called IA, issuing authority), and another opening group signatures (called OA, opening authority) ([6, 7, 8, 9, 10] etc.). The goal is to balance the burdens of servers and provide strong security against corrupted IA or OA.

Additionally a third party public key infrastructure can be deployed [12]. Any group member candidate also has a *user secret key* and a *user public key* for another signature scheme independent from the group signature scheme. When joining in the group, a group member candidate generates a signature on the protocol transcripts using his user secret key, GM stores the transcripts and signature in a database. When a dispute occurs later, e.g., a group member accuses GM of forging member certificate and group signatures for him, GM can rebut the accusation by showing the signature.

1.2 ID-based Group Signature

ID-based cryptosystems have a feature that encryption of plaintexts or verification of signatures do not have to be referred to a CA, i.e., Certificate Authority, for public keys. In the case of group signatures, either verification of group signature (VERIFY algorithm) or verification of output of "opening" (JUDGE algorithm), or both of them can be chosen to be identity based.

If only VERIFY is chosen to be identity based, as the first ID-based group signature [13] where verification was carried out by evaluating on all the identities of members in the group, the group signatures are not efficient enough.

If only JUDGE is chosen to be identity based, as in [14, 15, 16], we call it a *member ID-based group signature.* Unfortunately most of the schemes are found insecure[17, 18].

If both VERIFY and JUDGE are chosen to be identity based, i.e., group members and GM are all identity based, it is called a *fully identity based group signature* [19].

But the identity based JUDGE in [19] has a drawback of CA-frameability, i.e., it is frameable by dishonest CA (the CA may be different from GM), because member certificates are ID-based signatures from CA and all ID-based signatures are known to have the problem of key escrow.

The member ID-based group signature in [20] is a modification of ACTJ's group signature [6] as follows: the certificate (A_i, e_i) and secret key x_i of a member with identity ID_i satisfy $a^{x_i}a_0 = A_i^{H(ID_i)e_i} \bmod n$, where $H(ID_i)$ is a hash evaluation on identity string ID, instead of $a^{x_i}a_0 = A_i^{e_i} \bmod n$. The resulted group signature generation complexity and signature size are comparative to our proposal, but GM still has to remember the link between ID, $H(ID)$ and $h^{H(ID)}$, it is not a real member ID-based group signature in this sense.

In other words, there exists no efficient secure member ID-based group signature without CA-frameability yet, as far we know.

1.3 Model and Definition of (Member ID-based) Group Signature

The member ID-based group signature proposed in this paper is in line with the following definition and model, which is very similar to ordinary group signature [21, 12], except that real identity has replaced pseudo-name there.

Our member ID-based group signature defined as follows is also different from other ID-based schemes where extraction algorithms are required to generate secret keys from identities – where shortcoming of key escrow comes from.

Definition 1. A group signature is a signature scheme composed of the following algorithms between GM, members and verifiers.

- **SETUP:** an algorithm run by GM (IA and OA) to generate group public key gpk and group secret key gsk;
- **JOIN:** a probabilistic interactive protocol between GM (IA) and a group member candidate ID_i. If the protocol finishes successfully, the candidate becomes a new group member with member secret key msk_i and member certificate $cert_i$; and GM (IA) adds an entry for i (denoted as reg_i) in its registration table reg storing the protocol transcript, e.g. $cert_i$.
- **SIGN:** a probabilistic algorithm run by a group member, on input a message m and $msk_i, cert_i$, returns a group signature σ;
- **VERIFY:** a deterministic algorithm which, on input a message-signature pair (m, σ) and GM's public key gpk, returns 1 or 0 indicating the group signature is valid or invalid respectively;
- **OPEN:** a deterministic algorithm which, on input a message-signature pair (m, σ), secret key gsk of GM (OA), returns identity of the group member who signed the signature, and a proof π.
- **JUDGE:** a deterministic algorithm with output of OPEN as input, returns 1 or 0, i.e., the output of OPEN is valid or invalid.

Remark 1. The verification of outputs of OPEN is implicitly included in [21]. [12] explicitly defined it as an extra algorithm JUDGE .

We roughly describe a formal adversary model of group signature, the more formal definition is referred to [21]. Note that the definition here is a bit different from [21]. A major difference are that $\mathcal{O}_{read}$, $\mathcal{O}_{write}$, and the interface oracle state string $state_I$ are missing, the reasons are as follows: what $\mathcal{O}_{read}$ returns are $state_I$, which is useful only when $\mathcal{O}_{b_join}$ has been queried. In defining anonymity and traceability, $\mathcal{O}_{b_join}$ is never queried, so it is insignificant to query $\mathcal{O}_{read}$ in these scenarios. As for $\mathcal{O}_{write}$, what this oracle does is inserting to St(defined below). In defining non-frameability, GM is adversarially controlled, insertions are surely easy to do without having to refer to $\mathcal{O}_{write}$. We comment that these difference is only on the sense of description, and will not affect the security results.

The model specifies a series of oracles, some major oracles of them are described as follows:

Define a public-state string St, which is composed of St_{user} and St_{trans} that are both initialized to empty.

$\mathcal{O}_{pub}$: returns the group public key.

$\mathcal{O}_{key}$: returns the group secret key.

$\mathcal{O}_{a-join}$: An adversary might want to actively join in the group by controlling some members, that is query this oracle. This oracle will respond join requests from adversaries by acting as GM (IA). If the interaction is successful, the member and joining in transcript will be added into St_{user} and St_{trans} respectively. The user will also be marked as U^a (adversarially controlled).

$\mathcal{O}_{b-join}$: The adversarial GM might want to attack honest members during execution of JOIN, that is query this oracle. This oracle will act as honest group members interacting with GM. If the interaction is successful, the member and joining in transcript will be added into St_{user} and St_{trans} respectively. The user will also be marked as U^b (honest).

$\mathcal{O}_{sign}(ID_i, m)$: If group member ID_i is created from $\mathcal{O}_{b-join}$, this oracle returns a group signature of ID_i on m.

$\mathcal{O}_{open}(m, \sigma)$: This oracle returns the result of running OPEN on (m, σ), i.e., the identity of the group member and a proof of its claim.

$\mathcal{O}_{ch}(b, ID_{i_0}, ID_{i_1}, m)$: This oracle returns a valid group signature σ on m with ID_{i_b} being the signer.

Anonymous. Assume an adversary want to distinguish the signer of a group signatures from another group signature when GM is not adversarially controlled. It can be depicted as follows: the adversary is allowed to query oracles $\mathcal{O}_{pub}$, $\mathcal{O}_{a-join}$, $\mathcal{O}_{open}$ at most polynomial times. Then $\mathcal{A}$ query $\mathcal{O}_{ch}(b, ., ., .)$ with input (ID_{i_0}, ID_{i_1}, m). The challenge oracle $\mathcal{O}_{ch}(b, ., ., .)$ is expected to return a valid group signature σ on m by member i_b. Now $\mathcal{A}$ is still allowed to query the above oracles except that $\mathcal{O}_{open}(m, \sigma)$ is forbidden to query (denoted as $\mathcal{O}_{open}^{\neg\{\sigma\}}$). $\mathcal{A}$ should try to distinguish the two challenge oracles $(b = 0, 1)$. A group signature scheme is anonymous if the probability for any polynomial time bounded $\mathcal{A}$ to win is negligible, i.e., $\Pr[G^A_{anon-1}(1^v) = 1] - \Pr[G^A_{anon-0}(1^v) = 1]$ is negligible.

Experiment $G^A_{anon-b}(1^v)$, $b \in \{0, 1\}$
$(gpk, gsk) \leftarrow$ Setup (1^v);
$(aux, ID_{i_0}, ID_{i_1}, m) \leftarrow A^{\mathcal{O}_{pub}, \mathcal{O}_{a-join}, \mathcal{O}_{open}}$,
$(aux, \sigma) \leftarrow A^{\mathcal{O}_{Ch}(b,.,.,.)}$,
$d \leftarrow A^{\mathcal{O}_{pub}, \mathcal{O}_{a-join}, \mathcal{O}_{open}^{\neg\{\sigma\}}}(aux)$,
return d.

Table 1: Anonymity.

A detailed discuss of anonymity of group signature is presented in Section 1.3.1.

Traceable. Namely misidentification in [21]. Assume an adversary has controlled some group members, and they want to produce a valid group signature that would fail to trace to one of their identities when GM is not adversarially controlled. It can be depicted as follows: adversary $\mathcal{A}$ is allowed to query oracles $\mathcal{O}_{pub}$, $\mathcal{O}_{a-join}$, $\mathcal{O}_{open}$ at most polynomial times, and it should output a message signature pair (m, σ). $\mathcal{A}$ wins if σ is valid and $OPEN(m, \sigma, gsk)$ fails to output members in U^a. A group signature scheme is traceable if the probability for any polynomial time bounded $\mathcal{A}$ to win is negligible, i.e., $\Pr[G^A_{trace}(1^v) = 1]$ is negligible.

```
Experiment $G^A_{trace}(1^v)$
$(gpk, gsk) \leftarrow$ Setup $(1^v)$, $r \leftarrow 0$,
$(m, \sigma) \leftarrow A^{\mathcal{O}_{pub}, \mathcal{O}_{a-join}, \mathcal{O}_{open}}$,
if VERIFY$(m, \sigma, gpk) = 1$,
    $(ID_i, \pi) \leftarrow$ OPEN$(m, \sigma, gsk)$,
    if JUDGE$(m, \sigma, ID_i, \pi) = 1$ and $ID_i \notin U^a$,
    $r \leftarrow 1$.
return $r$.
```

Table 2: Traceability.

Non-frameable. Assume an adversary want to produce a valid group signature that would trace to an honest member when GM is adversarially controlled. It can be depicted as follows: adversary $\mathcal{A}$ is allowed to query $\mathcal{O}_{pub}$, $\mathcal{O}_{key}$, $\mathcal{O}_{b-join}$, $\mathcal{O}_{sign}$ at most polynomial times,it should output a message signature pair (m, σ). $\mathcal{A}$ wins if σ is valid and $OPEN(m, \sigma, gsk)$ fails to output members in U^b, and $\mathcal{O}_{sign}(., m)$ has not been queried to this honest member (denoted as $(ID_i, m) \notin$ hist$(\mathcal{O}_{sign})$). A group signature scheme is non-frameable if the probability for any polynomial time bounded $\mathcal{A}$ to win is negligible, i.e., $\Pr[G^A_{frame}(1^v) = 1]$ is negligible.

```
Experiment $G^A_{frame}(1^v)$
$(gpk, gsk) \leftarrow$ Setup $(1^v)$, $r \leftarrow 0$,
$(m, \sigma) \leftarrow A^{\mathcal{O}_{pub}, \mathcal{O}_{key}, \mathcal{O}_{b-join}, \mathcal{O}_{sign}}$,
if VERIFY$(m, \sigma, gpk) = 1$,
    $(ID_i, \pi) \leftarrow$ OPEN$(m, \sigma, gsk)$,
    if JUDGE$(m, \sigma, ID_i, \pi) = 1$ and $ID_i \in U^b$ and $(ID_i, m) \notin$ hist$(\mathcal{O}_{sign})$,
    $r \leftarrow 1$.
return $r$.
```

Table 3: Non-frameability.

Definition 2. A group signature scheme is secure if it is anonymous, traceable and non-frameable.

In the definition above, there exists a risk that GM might be corrupted, e.g., GM can frame an honest group member ID_u just by selecting a new member secret key msk_u and generating a new member certificate $cert_u$, any group signature generated from $(msk_u, cert_u)$ is valid and can be opened to reveal ID_u.

This can be fixed by introducing an additional trusted third authority CA independent from GM as explicitly defined in the model of [12]: every member is given a user public key from CA and a user secret key kept to himself; a member sends a commitment to its member secret key, along with a signature of it using his user secret key, to GM (IA); GM (IA) generates a member certificate on the commitment; a group signature should include an encryption of the commitment that has been signed; execution of OPEN should reveal the commitment and the signer identity is obtained by matching it with stored transcript; the group member will not be able to repudiate because he has signed on this commitment which is now correctly decrypted from the group signature. Note that this method is applicable to all group signature schemes, the application to our proposal is easier (Section 3.5.2), so we will not consider such GM corruption risk in the sequel and assume that GM is trusted, since otherwise we can make it trusted by an independent CA.

1.3.1 A Discussion of Anonymity

The provability of ACJT's scheme in the formal model [12] has been questioned in [10]. The reason is mainly due to the IND-CPA secure ElGamal encryption [22] adopted in ACJT's scheme. After enhancing ElGamal into an IND-CCA2 secure scheme, the enhanced ACJT's scheme is formally proved secure in another formal model [21] which is subtly different from [12]. We point out that this is a misunderstanding that an IND-CCA2 secure encryption scheme is necessary to compose a secure group signature.

Two Different Oracles. The problem lies in a confusion between open oracle $\mathcal{O}_{open}$ in a group signature and decryption oracle $\mathcal{O}_{dec}$ in the underlying encryption scheme in security proofs. Anonymity of ACJT is often reduced to indistinguishability of ElGamal, any query to $\mathcal{O}_{open}$ is intercepted and the encryption part is extracted and transferred to $\mathcal{O}_{dec}$. But in OPEN algorithm, the first step is to verify the validity of a group signature. $\mathcal{O}_{open}$ could and should only allow valid group signatures in, and transfer their encryption parts to $\mathcal{O}_{dec}$. If this sounds familiar, that is the way we construct IND-CCA2 encryption scheme from IND-CPA scheme [23].

Decryption (Open) Oracle can be different. We classify a $\mathcal{O}_{dec}$ ($\mathcal{O}_{open}$) oracle in two levels:

– Responsible Oracle: Given a purported ciphertext (group signature), a responsible $\mathcal{O}_{dec}$ ($\mathcal{O}_{open}$) will never apply its private decryption key until it is assured of the validity of the ciphertext (group signature).

– Irresponsible Oracle: Given a purported ciphertext (group signature), an irresponsible $\mathcal{O}_{dec}$ ($\mathcal{O}_{open}$) always inadvertently apply its private decryption key without caring for the validity of the ciphertext (group signature).

Irresponsible decryption oracle reflects a not well designed decryption software that might misuse its decryption key by decrypting whatever it has got before checking the validity of the ciphertext, throw away decryption outputs inadvertently when they are found meaningless. In the sequel, we assume that all $\mathcal{O}_{dec}$ and $\mathcal{O}_{open}$ oracles are responsible.

The classification and assumption are reasonable. It is well known that an IND-CCA2 encryption scheme is available by double encrypting the same message under an IND-CPA encryption scheme [24], which is exactly the method used to enhance ACJT's scheme, e.g. [21, 10]. The resulting IND-CCA2 ciphertext consists of two independent IND-CPA ciphertexts and a proof that the same plaintext is encrypted. The strong security of IND-CCA2 comes from the difficulty of composing valid ciphertexts from the challenged cipher by a computation bounded adversary, and the decryption oracle is assumed to decrypt valid ciphertexts only. Here the decryption oracle has already been implicitly assumed responsible, because an irresponsible decryption oracle can decrypt either one of the two ciphertexts even when the whole ciphertext is invalid. So even the encryption enhanced group signatures, e.g., [21, 10], may be no longer secure if $\mathcal{O}_{dec}$ is irresponsible. Thus the assumption of responsible $\mathcal{O}_{open}$ is reasonable.

A further discussion is referred to [25].

1.4 Our Contribution

We propose an interesting efficient member ID-based group signatures, i.e., verification of output from algorithm OPEN does not have to refer to a registration table (acting as certification list).

The proposal is CA (GM in the case of group signature) non-frameable, i.e., secret key of member is not escrowed to GM, which is unique among all known member ID-based group signatures as far as we know.

Extra Feature 1. In the proposal, GM does not have to maintain a registration table to obtain the real identity of the signer. We observe that in most group signature schemes [3, 4, 5, 6, 7, 8, 9, 10], GM firstly derives some information from the group signature, then searches for the identity of signer in the registration table.

The significance of eliminating the registration table and the reason why the methods attaining the goal are not straightforward are as follows.

There may be arguments of the motivation for eliminating the registration table. Some may feel it natural to have a table to check and deem its elimination unnecessary. That is because we have seen none of the group signatures without doing so. Why should we distrust the output of OPEN and refer to the registration table for consistency, while we would trust every piece of plain-text obtained from a decryption algorithm and never bother to refer to somewhere for a checkup?

A natural approach to free GM from retrieving the registration database may be let GM include the identity of the group member in the member certificate, but generally this will not work, because another problem will arise: how to generate the group signature efficiently? Since current efficient group signatures are in fact proof of knowledge of membership certificate and user secret key.

The compact group signature in [26] which has an advantage of provability without random oracles, also enables GM or tracer to retrieve signer identity directly. But it has a disadvantage that its signing time, verification time, and signature size are all logarithmic in the number of signers. Another shortcoming of [26] is that it permits GM-frameability, i.e., the whole group signing key of every group member is known to GM and GM is able to generate group signatures in the name of any group member.

Remark 2. The above arguments are on the condition that GM is not corrupted and trustable which is so in most cases. In case of corrupted GM or GM is not trustable, registration table is unavoidable, please refer to the end of Section 1.3 and Section 3.5.2 for a detailed discussion.

Extra Feature 2. In case registration table is preferred, the proposal provides an alternative countermeasure against tampered registration table to applying integrity techniques to the table: the honest GM firstly opens a group signature and obtains the identity of a group member directly, then GM can choose to continue searching the identity in the registration table; if they are same, it is ok, otherwise GM can conclude that either the registration table has been tampered by someone, or the group signature scheme has been broken. But in previous group signature schemes, GM has to completely rely on stored registration table to retrieve identities, and any inconsistence with registration table will lead to failure of OPEN.

Extra Feature 3. Our proposal is the first application of partial trapdoor one-way functions [27] as far as we know.

Efficiency. Besides all the above benefits, the proposal is efficient. The signing time, verification time, and signature size (2.3 times length of [6]) are all constant, independent with the number of signers.

2 Notations and Preliminary Background

2.1 Notations

Notations appear in the sequel are defined as follows.

- $PK\{(\alpha, \beta, ...) : R(\alpha, \beta, ...)\}$, denotes Proof of Knowledge, an interactive protocol of proving knowledge of $(\alpha, \beta, ...)$ satisfying relation $R(\alpha, \beta, ...)$.

- $SK\{(\alpha, \beta, ...) : R(\alpha, \beta, ...)\}\{m\}$, denotes Signature of Knowledge, a non-interactive protocol of proving knowledge of $(\alpha, \beta, ...)$ satisfying relation $R(\alpha, \beta, ...)$.
- $\pm\{0,1\}^k$, a set of k bits long integers.
- $H(a, b, c, ..., m)$, a hash function evaluated on concatenate of $a, b, c, ..., m$.
- $S(2^{l_x}, 2^{\mu_x})$ denotes integer interval $(2^{l_x} - 2^{\mu_x}, 2^{l_x} + 2^{\mu_x})$.

2.2 Paillier's Cryptosystem[27]

The public key cryptosystem is defined in $Z^*_{n^2}$.

Let $n = pq$, $\phi(n) = (p-1)(q-1)$, $\lambda(n) = lcm(p-1, q-1)$.

Structure of $Z^*_{n^2}$. The order of $Z^*_{n^2}$ is $n\phi(n)$ and $Z^*_{n^2} \backsimeq Z_n \times Z^*_n$. There exists isomorphic maps ϵ_g from $Z_n \times Z^*_n$ to $Z^*_{n^2}$: $\epsilon_g(x, y) = g^x y^n \bmod n^2$, given any $g \in Z^*_{n^2}$ with order multiple of n.

Partial Discrete Logarithm Problem. $PDL[n, g]$ is defined as follows: given $w \in \langle g \rangle$, compute $x \in Z_n$, denoted as $[w]_g$, that $\epsilon_g(x, y) = w \bmod n^2$, y can be any value in Z^*_n.

Suppose the order of g is αn $(1 \leq \alpha \leq \lambda)$, $PDL[n, g]$ is easy to compute given trapdoor α or the factorization of n:

$$[w]_g = \frac{L(w^\alpha \bmod n^2)}{L(g^\alpha \bmod n^2)} \bmod n,$$

where L is defined as: $L(u) = (u-1)/n, u \in Z^*_{n^2}$ and $u = 1 \bmod n$. Otherwise, $PDL[n, g]$ is assumed hard when unknown α is not small. So PDL problem is assumed hard with trapdoor.

Discrete Logarithm Problem over $Z^*_{n^2}$. $DL[n, g]$ is defined as follows: given $w \in \langle g \rangle$, compute x, that $g^x = w \bmod n^2$. DL problem over $Z^*_{n^2}$ is assumed hard without trapdoor.

Setting $\alpha = \lambda/2$ will be the case discussed in [28]. Now $\langle g \rangle = QR_{n^2}$, the cyclic group of quadratic residues modulo n^2.

Decisional Diffie-Hellman Problem over $Z^*_{n^2}$. In group QR_{n^2}, where $n = pq = (2p'+1)(2q'+1)$, DDH problem over $Z^*_{n^2}$ is to distinguish between $g^{xy} \bmod n^2$ and $g^z \bmod n^2$, given $g^x \bmod n^2$ and $g^y \bmod n^2$ for unknown random $x, y, z \in [1, pqp'q']$. DDH problem over $Z^*_{n^2}$ is assumed hard when factorization of n is unknown [28].

Strong RSA Problem over $Z^*_{n^2}$. Strong RSA problem is hard over $Z^*_{n^2}$ if strong RSA assumption hold in Z^*_n, it is self-evident that if (u, v) can be computed for a randomly chosen $y \in Z^*_{n^2}$ satisfying $u^v = y \bmod n^2$, then $u^v = y \bmod n$ meaning Strong RSA solved for y in Z^*_n.

Partial Trapdoor One-way Function. It is well known that in a large prime ordered group G, $f(x) = g^x$ is a one-way function without trapdoor; and in group Z^*_n (n is an integer with two large prime factors),

$RSA(y) = y^e \bmod n$ is a trapdoor one-way function meaning that y is completely computable in polynomial time given factorization of n.

In $Z^*_{n^2}$, however, $f(x) = g^x \bmod n^2$ has a property somewhere between the two kinds of one-way functions, that is partial trapdoor one-way function, which means only part of preimage is polynomial computable even if trapdoor is given.

The partial trapdoor one-way function has not been fully utilized in any application. Paillier's Cryptosystem [27] only took the advantage of the property as a normal trapdoor one-way function. In Section 3, we will show its first application in group signatures.

2.3 ACJT's Group Signature[6].

Ateniese et al. proposed the first practical group signature based on strong RSA assumption. Later it is improved and proved in a formal model [21].

It begins by choosing proper security parameters that satisfy the following relationships: a collision resistant hash function H, two intervals $\Gamma = S(2^{l_e}, 2^{\mu_e}) \subseteq \{1, ..., 2^{2l_p-2}\}$, $\Delta = S(2^{l_x}, 2^{\mu_x})$, where $l_x > \epsilon(\mu_x + k) + 2$, $l_e > \epsilon(\mu_e + k) + 2, \mu_e > l_x + 2$. ϵ is any real number larger than 1. $l_e, \mu_e, l_x, \mu_x, l_p, k$ are integers. Note that Γ and Δ are two distant disjoint intervals.

SETUP. IA randomly chooses two l_p bits long safe primes p, q, i.e, $p' = (p-1)/2$ and $q' = (q-1)/2$ are large primes too and $a, a_0, g, h \in_R QR_n$ where $n = pq$, IA's secret key is $\{p', q', p, q\}$; OA chooses secret key $x \in_R Z_n$, calculates $y = g^x \bmod n$. Group public key is $gpk = \{n, a, a_0, y, g, h\}$.

JOIN. Group member candidate U_i randomly selects his member secret key $x_i \in \Delta$ and sends a^{x_i} as well as a proof π_i of knowledge of $x_i \in \Delta$ to IA; IA randomly chooses a prime $e_i \in_R \Gamma$, calculates $A_i := (a^{x_i} a_0)^{1/e_i} \bmod n$, and sends member certificate (A_i, e_i) to U_i. IA sets $reg_i = (A_i, e_i, a^{x_i}, \pi_i, U_i)$. In the end U_i's group signing key is (A_i, e_i, x_i).

SIGN and VERIFY. U_i signs on m by computing $T_1 = A_i y^r \bmod n$, $T_2 = g^r \bmod n$, $T_3 = g^{e_i} h^r \bmod n$ and generating an honest verifier zero-knowledge proof of $(A_i, e_i \in \Gamma, x_i \in \Delta)$, which is formulated specifically as follows

$$\begin{aligned}\tau = SK\{(\alpha, \beta, \gamma, \delta) : a_0 = T_1^\alpha / a^\beta y^\gamma \bmod n, T_2 = g^\delta \bmod n,\\ 1 = T_2^\alpha / g^\gamma \bmod n, T_3 = g^\alpha h^\delta \bmod n, \alpha \in \Gamma, \beta \in \Delta\}\{m\},\end{aligned}$$

The verification of the group signature $\sigma = (T_1, T_2, T_3, \tau)$ is the verification of the above proof.

OPEN. Given a group signature $\sigma = (T_1, T_2, T_3, \tau)$ of m, OA firstly run VERIFY on it, if it is valid then calculates $A := T_1 / T_2^x \bmod n$, compares it with items in reg, if there is some j that $A \in reg_j$, OA concludes that the signer is $User_j$, outputs $(j, \tau' = SK\{x : y = g^x \bmod n, T_1/A_j = T_2^x \bmod n\}, reg_j)$.

JUDGE. Given (m, σ, j, τ'), run the verification of τ', output 1 if it is valid, 0 otherwise .

The following Lemmas has been proved in [6].

Lemma 2.1. *The above scheme is coalition resistant, i.e., given polynomial number of (A_i, e_i, x_i), it is negligible for an polynomial time bounded adversary to forge a new (A^*, e^*, x^*), assuming Strong RSA assumption on QR_n.*

Lemma 2.2. *The interactive protocol underlying the above scheme (SIGN and VERIFY) is statistical honest verifier zero-knowledge and sound, under Strong RSA assumption over QR_n.*

It was further proved [21] that

Theorem 2.3. *The above scheme is secure against misidentification attacks, i.e., traceable, assuming Strong-RSA problem is hard over QR_n, in random oracle model.*

Theorem 2.4. *The above scheme is non-frameable assuming Discrete-logarithm problem is hard over QR_n with known factorization, in random oracle model.*

Theorem 2.5. *The above scheme is anonymous, assuming DDH with known factorization is hard over QR_n, in random oracle model.*

Remark. The group signature scheme described here is a bit different from [21] that A_i is single ElGamal encrypted here, but A_i is double ElGamal encrypted there: $(A_i y^r, g^r, A_i y'^{r'}, g'^{r'})$ and a proof of the same A_i is encrypted in two ciphertexts. As discussed in Section 1.3.1, single ElGamal encryption is enough under responsible $\mathcal{O}_{open}$.

3 The Proposal of Member ID-based Group Signature

3.1 Brief Idea

The idea is to apply [6] on a different group, i.e., QR_{n^2}, where a partial trapdoor one-way function exists [27]. Following the same idea, more such group signatures are obtained from [7, 29, 30]. It is not a straightforward conversion or combination, there are some tricks in OPEN algorithm.

Recall that in ACJT's original scheme, the member certificate (A_i, e_i) and member secret key x_i satisfy $A_i^{e_i} = a^{x_i} a_0 \bmod n$, adding in identity and changing the underlying group to QR_{n^2} will get $A_i^{e_i} = a^{ID_i + nx_i} a_0 \bmod n^2$. The JOIN, SIGN and VERIFY are quite similar to original ACJT scheme, except that $a^{ID_i+nx_i}$ is also encrypted besides A_i, OPEN can now get ID_i directly by solving PDL problem with trapdoor.

Note that our proposal description excludes the third party PKI for simplicity reason.

3.2 The Proposal

SETUP. Choose a l_n bits long RSA modulo $n = pq = (2p'+1)(2q'+1)$, let QR_{n^2}, a cyclic group with order $pqp'q'$, denotes the quadratic residue in $Z^*_{n^2}$. Further choose the following parameters:

$H : \{0,1\}^* \mapsto \{0,1\}^k$: a collision resistant hash function with a proper security parameter k;

l_x, μ_x: $S(2^{l_x}, 2^{\mu_x}) \subseteq (1, p'q')$, i.e., interval $(2^{l_x} - 2^{\mu_x}, 2^{l_x} + 2^{\mu_x}) \subseteq (1, p'q')$;

l_z, μ_z: $S(2^{l_x}, 2^{\mu_x}) \subseteq (\frac{2^{l_z}-2^{\mu_z}}{n}, \frac{2^{l_z}+2^{\mu_z}}{n} - 1)$, $l_z > \epsilon(k + \mu_z) + 2$;

ϵ: any real number greater than 1;

l_e, μ_e: $l_e > \epsilon(k + \mu_e) + 2$, $\mu_e > l_z + 2$.

Group secret key is $(x \in_R Z^*_{pqp'q'}, p, q, p'q')$.

Group public key is $gpk = (n, a_0, a, g, h, y,\ l, \mu, l_e, \mu_e, l_z, \mu_z, \epsilon)$, where (a_0, a, g, h) are random generators of QR_{n^2}, $y = g^x \bmod n^2$.

Let $\Gamma = S(2^{l_e}, 2^{\mu_e})$, $\Delta = S(2^{l_z}, 2^{\mu_z})$.

JOIN. A user with identity $ID_i \in Z^*_n$ becomes the a group member in the following steps after GM has authenticated that it is really talking with ID_i.

$ID_i \rightarrow$ GM: ID_i selects $x_i \in_R S(2^{l_x}, 2^{\mu_x})$, computes $z_i = ID_i + nx_i$, $C_i = a^{z_i} \bmod n^2$, sends C_i, and a proof of knowledge of $x_i \in_R S(2^{l_x}, 2^{\mu_x})$ that $C_i a^{-ID_i} = (a^n)^{x_i}$;

$ID_i \leftarrow$ GM: GM chooses a prime $e_i \in S(2^{l_e}, 2^{\mu_e})$, computes $A_i = (a_0 C_i)^{\frac{1}{e_i}} \bmod n^2$, and sends them to ID_i.

ID_i checks that the member certificate (A_i, e_i) and member secret key x_i satisfy $a^{ID_i+nx_i} a_0 = {A_i}^{e_i} \bmod n^2$.

SIGN. Similar to ACJT scheme except that the computations are in $Z^*_{n^2}$ and a^{z_i}, where $z_i = ID_i + nx_i$, is encrypted instead of A_i. The details are as follows:

(1) To sign a message m, ID_i encrypts a^{z_i} under GM's public key y, and randomize A_i: $W_1 = a^{z_i} y^r \bmod n^2$, $W_2 = g^r \bmod n^2$, $W_3 = A_i^r \bmod n^2$, $W_4 = g^{e_i} h^r \bmod n^2$, where $r \in_R \{0,1\}^{2l_n-2}$.

(2) ID_i then generates a signature of knowledge:

$$SK\{(e_i, z_i, r, rz_i) : W_1 = a^{z_i} y^r \bmod n^2, W_2 = g^r \bmod n^2, 1 = a^{rz_i} a_0^r W_3^{-e_i} \bmod n^2,$$
$$1 = g^{rz_i} W_2^{z_i} \bmod n^2, W_4 = g^{e_i} h^r \bmod n^2, e_i \in S(2^{l_e}, 2^{\mu_e}), z_i \in S(2^{l_z}, 2^{\mu_z})\}\{m\}.$$

that can be realized as follows:

(2.1) Choose $k_1 \in_R \pm\{0,1\}^{\epsilon(k+\mu_e)}$, $k_2 \in_R \pm\{0,1\}^{\epsilon(k+\mu_z)}$, $k_3 \in_R \pm\{0,1\}^{\epsilon(k+2l_n-2)}$, $k_4 \in_R \pm\{0,1\}^{\epsilon(k+2l_n+l_z-2)}$, and compute

$$R_1 = a^{k_2} y^{k_3} \bmod n^2, \quad R_2 = g^{k_3} \bmod n^2, \quad R_3 = a^{k_4} a_0^{k_3} W_3^{-k_1} \bmod n^2,$$
$$R_4 = g^{k_4} W_2^{-k_2} \bmod n^2, \quad R_5 = g^{k_1} h^{k_3} \bmod n^2.$$

(2.2) Calculate $c = H(gpk, m, W_1, W_2, W_3, W_4, R_1, R_2, R_3, R_4, R_5)$ and

$$s_1 = k_1 - c(e_i - 2^{l_e}), \quad s_2 = k_2 - c(z_i - 2^{l_z}), \quad s_3 = k_3 - cr,$$
$$s_4 = k_4 - crz_i \text{ (in Z)}.$$

(2.3) The group signature on m is $\sigma = (W_1, W_2, W_3, W_4, s_1, s_2, s_3, s_4, c)$.

VERIFY. Given $\sigma = (W_1, W_2, W_3, W_4,\ s_1, s_2, s_3, s_4, c)$ as a purported group signature on m, verifiers check if

$$c = H(gpk, m, W_1, W_2, W_3, W_4, W_1^c a^{s_2 - c2^{l_e}} y^{s_3} \bmod n^2, g^{s_3} W_2^c \bmod n^2,$$
$$a^{s_4} a_0^{s_3} W_3^{-(s_1 - c2^{l_e})} \bmod n^2, g^{s_4} W_2^{-s_2+c2^{l_z}} \bmod n^2, g^{s_1 - c2^{l_e}} h^{s_3} W_4^c \bmod n^2)$$

and $s_1 \in \pm\{0,1\}^{\epsilon(k+\mu_e)+1}$, $s_2 \in \pm\{0,1\}^{\epsilon(k+\mu_z)+1}$, $s_3 \in \pm\{0,1\}^{\epsilon(k+2l_n-2)+1}$, $s_4 \in \pm\{0,1\}^{\epsilon(k+2l_n+l_z-2)+1}$.

The group signature is valid if the above requirements are satisfied, invalid otherwise.

OPEN. Given a group signature, GM firstly verifies its validity, decrypts (W_1, W_2) to get $a^{ID_i+nx_i} = W_1/W_2{}^x \bmod n^2$ if that is the case, then further decrypts $a^{ID_i+nx_i}$ to get ID_i as solving PDL problem in QR_{n^2} (Section 2.2). GM also outputs two non-interactive zero-knowledge proofs $\tau_1 = SK_1\{x : y = g^x \bmod n^2, W_2^x = W_1/D \bmod n^2\}$ and $\tau_2 = SK_2\{\gamma : (a^{ID_i}/D)^\gamma = 1 \bmod n^2, (a^n)^\gamma = 1 \bmod n^2\}$, where $D = a^{ID_i+nx_i}$. The output is $(ID_i, D, \tau_1, \tau_2)$.

JUDGE. The output of OPEN is verified by checking τ_1 and τ_2.

A possible set of parameters is $\epsilon = 1.1$, $l_n = 1024$, $\mu_z = 1623$, $l_z = 1963$, $\mu_x = 598$, $l_x = 939$, $\mu_e = 1965$, $l_e = 2339$, $k = 160$ and the group signature length will be 19669 bits, i.e., 2458 bytes, about 2.3 times of that of [6] under parameters $\epsilon = 1.1$, $l_n = 1024$, $k = 160$, $l_x = 838$, $\mu_x = 600$, $l_e = 1102$ and $\mu_e = 840$ [10].

3.3 Other Member ID-based Group Signatures

Applying our idea to existing group signatures based on Strong RSA assumption, e.g. [7, 29, 30], will also lead to member ID-based group signatures that have similar features to the above proposal.

Take [7] as an example, the member certificate (r_i, s_i) and member secret key x_i satisfy $g^{x_i} = r_i y_1^{r_i} g_1^{s_i} \bmod n$ in original scheme, they will satisfy $g^{ID_i+nx_i} = r_i y_1^{r_i} g_1^{s_i} \bmod n^2$ instead after our idea being applied. The security analysis is similar to Section 3.4.

3.4 Security Proofs

Lemma 3.1. *The interactive protocol underlying the proposal (SIGN and VERIFY) is statistical honest verifier zero-knowledge and sound, under Strong RSA assumption over QR_{n^2}.*

Proof. See Appendix A. □

Theorem 3.2. *The proposal is traceable under Strong RSA assumption over QR_{n^2}, in random oracle model.*

Proof. See Appendix B. □

Theorem 3.3. *On the condition of Lemma 3.1, the proposal is anonymous against adversaries except IA, i.e., GM in this scheme, under DDH assumption over QR_{n^2} when factor of n is unknown, in random oracle model.*

Proof. See Appendix C. □

Theorem 3.4. *The proposal is non-frameable against adversaries (including GM) under Partial Discrete Logarithm assumption over QR_{n^2}, in random oracle model.*

Proof. See Appendix D. □

3.5 Discussions

3.5.1 Revocation

The efficient revocation method provided for ACJT's scheme [31, 8] is still applicable to our proposals. GM lets e_{ID} be the $R(ID)$-th prime in Γ, where R is a pseudo random function. To revoke ID from the group, GM just recalculates and publishes e_{ID} and updates the dynamic accumulator, i.e., sets $a \leftarrow a^{\frac{1}{e_{ID}}}$, $a_0 \leftarrow a_0^{\frac{1}{e_{ID}}}$.

For the original ACJT scheme, the revocation method above has a disadvantage that running OPEN algorithm on a group signature with encrypted certificate, i.e., A_i, might result in a different value from the stored certificates by GM, because the members might have updated their certificates since they were firstly issued member certificates. Although this problem can be fixed in a few ways, our proposal inherently overcomes this disadvantage because no matter how the member certificates have been updated, decrypting a group signature always outputs the identity string.

3.5.2 Countermeasure for Corrupted GM

To counter the risk of corruption of GM mentioned in Section 1.3, modify the proposal as follows:

Each group member candidate with identity ID_i (e.g., social security number, registered name), selects his user secret key usk_i and obtains corresponding user public key upk_i in advance. He also chooses a member secret key $sk_i = x_i \in_R S(2^{l_x}, 2^{\mu_x})$, commits it to $Com_i = a^{x_i} \bmod n^2$, publishes reg_i=$(ID_i, Com_i, Sig_{usk_i}(Com_i))$ before joining a group. reg_i can be chosen not to publish, but it must be available to GM(OA) and judgers on disputes.

Given (ID_j, τ_1, τ_2) from OA, a judger checks wether τ_1 is $SK_1\{x : y = g^x \bmod n^2, W_2^x = W_1/(a^{ID_j} Com_j^n) \bmod n^2\}$ and wether τ_2 is $SK_2\{\gamma : [a^{ID_j}/(a^{ID_j} Com_j^n)]^\gamma = 1 \bmod n^2, (a^n)^\gamma = 1 \bmod n^2\}$.

reg here only needs to store a commitment of secret key sk_{ID} and a signature on it, they are used to provide non-repudiation of the participation of the group signature generation, not for retrieving ID in OPEN as other group signature schemes [6, 12] etc..

reg here can be maintained by a third CA and generated even before a group member candidate joins in the group.

If non-repudiation is not required, i.e., GM is trusted that it will not generate a member certificate for any group member candidate without his awareness and participation, then reg can be totally discarded, GM does not need to remember anything except generated random numbers to prevent certificate collision among group members, actually even these are not necessarily to remember if pseudo-random functions are adopted.

4 Conclusions

We propose an efficient GM non-frameable member ID-based group signatures, i.e., verification of output from algorithm OPEN does not have to refer to a registration table (acting as certification list), and secret key of member is not escrowed to GM (which is unique among all known member ID-based group signatures as far as we know).

The proposal also has three extra features. The first is that GM does not have to maintain a registration table to obtain the real identity of the signer. The second is that it provides an alternative countermeasure against tampered registration table to applying integrity techniques to the table in case the registration table is preferred. The third is that it is the first application of partial trapdoor one-way functions [27] as far as we know.

References

[1] D. Chaum and E. van Heyst, "Group signatures," in *EUROCRYPT'91*, LNCS 547, pp. 257–265, Springer-Verlag, 1991.

[2] J. Camenisch, "Efficient and generalized group signatures," in *EUROCRYPT'97*, LNCS 1233, pp. 465–479, Springer, 1997.

[3] J. Camenisch and M. Stadler, "Efficient group signatures schemes for large groups," in *CRYPTO'97*, LNCS 1296, pp. 410–424, Springer-Verlag, 1997.

[4] J. Camenish and M. Michels, "A group signature scheme with improved efficiency," in *ASIACRYPT'98*, LNCS 1514, pp. 160–174, Springer, 1998.

[5] J. Camenisch and M. Michels, "A group signature scheme based on an RSA-variant," in *Technical Report RS-98-27. BRICS, University of Aarhus*, Primary version of this paper appeared at ASIACRYPT'98, Springer-Verlag, November 1998.

[6] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," in *CRYPTO'00*, LNCS 1880, pp. 255–270, Springer-Verlag, 2000.

[7] G. Ateniese and B. de Medeiros, "Efficient group signatures without trapdoors," in *ASIACRYPT'03*, LNCS 2894, pp. 246–268, Springer-Verlag, 2003.

[8] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *CRYPTO'04*, LNCS 3152, pp. 45–55, Springer-Verlag, 2004.

[9] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *CRYPTO'04*, LNCS 3152, pp. 56–72, Springer-Verlag, 2004.

[10] L. Nguyen and R. Safavi-Naini, "Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings," in *ASIACRYPT'04*, LNCS 3329, pp. 372–386, Springer-Verlag, 2004.

[11] A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," in *CRYPTO'86*, LNCS 263, pp. 186–194, Springer, 1987.

[12] M. Bellare, H. Shi, and C. Zhang, "Foundations of group signatures: The case of dynamic groups," in *CT-RSA'05*, LNCS 3376, pp. 136–153, Springer-Verlag, 2005. Full Paper at http://www-cse.ucsd.edu/ mihir/papers/dgs.html.

[13] S. Park, S. Kim, and D. Won, “Id-based group signature,” in *Electronics Letters*, 33(19), pp. 1616–1617, IEE, 1997.

[14] Y. Tseng and J. Jan, “A novel ID-based group signature,” in *Information Sciences*, 120, pp. 131–141, Elsevier Science, 1999.

[15] S.Xia and J.You, “A group signature scheme with strong separability,” in *The Journal of Systems and Software*, 60(3), pp. 177–182, Elsevier Science, 2002.

[16] S. Han, J. Wang, and W. Liu, “An efficient identity-based group signature scheme over elliptic curves,” in *ECUMN'04*, LNCS 3262, pp. 417–429, Springer-Verlag, 2004.

[17] M. Joye, S. Kim, and N.-Y. Lee, “Cryptanalysis of two group signature schemes,” in *Information Security*, LNCS 1729, pp. 271–275, Springer-Verlag, 1999.

[18] G. Wang, “Security analysis of several group signature schemes,” in *INDOCRYPT'03*, LNCS 2904, pp. 252–265, Springer-Verlag, 2003.

[19] V. K. Wei, T. H. Yuen, and F. Zhang, “Group signature where group manager, members and open authority are Identity-based,” in *ACISP 2005*, LNCS 3574, pp. 468–480, Springer-Verlag, 2005.

[20] Z. Chen, J. Huang, D. Huang, J. Zhang, and Y. Wang, “Provably secure and ID-based group signature scheme,” in *AINA'04*, vol. 02, pp. 384–387, IEEE, 2004.

[21] A. Kiayias and M. Yung, “Group signatures: Provable security, efficient constructions and anonymity from trapdoor-holders,” in *Cryptology ePrint Archive*, Report 2004/076, 2004.

[22] T. E. Gamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” in *CRYPTO'84*, LNCS 196, pp. 10–18, Springer, 1985.

[23] R. Cramer and V. Shoup, “Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack,” *SIAM J. Comput.*, vol. 33, no. 1, pp. 167–226, 2004.

[24] M. Naor and M. Yung, “Public-key cryptosystems provably secure against chosen ciphertext attacks,” in *22nd Annual ACM Symposium on the Theory of Computing*, pp. 427–437, ACM Press, 1990.

[25] S. Zhou and D. Lin, “On anonymity of group signatures,” in *CIS 2005, Part II*, LNAI 3802, pp. 131–136, Springer Verlag, 2005. Full paper in Cryptology ePrint Archive, Report 2005/422.

[26] X. Boyen and B. Waters, “Compact group signatures without random oracles.” Cryptology ePrint Archive, Report 2005/381, 2005.

[27] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in *EUROCRYPT’99*, LNCS 1592, pp. 223–238, Springer-Verlag, 1999.

[28] E. Bresson, D. Catalano, and D. Pointcheval, “A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications,” in *ASIACRYPT’03*, LNCS 2894, pp. 37–54, Springer-Verlag, 2003.

[29] J. Camenish and A. Lysyanskaya, “A signature scheme with efficient protocols,” in *SCN’02*, LNCS 2576, pp. 274–295, Springer, 2003.

[30] J. Camenisch and J. Groth, “Group signatures: Better efficiency and new theoretical aspects,” in *Security in Communication Networks: 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004*, LNCS 3352, pp. 120–133, Springer-Verlag, 2005.

[31] J. Camenisch and A. Lysyanskaya, “Dynamic accumulators and application to efficient revocation of anonymous credentials,” in *CRYPTO’02*, LNCS 2442, pp. 61–76, Springer-Verlag, 2002.

[32] I. Damgård and E. Fujisaki, “An integer commitment scheme based on groups with hidden order,” 2001.

[33] A. Kiayias, Y. Tsiounis, and M. Yung, “Traceable signatures,” in *EUROCRYPT’04*, LNCS 3027, pp. 571–589, Springer, 2004.

A Proof of Lemma 3.1

Proof. **Soundness:** By resetting the prover under the same random inputs, an honest verifier can get $(W_1, W_2, W_3, W_4, s_1, s_2, s_3, s_4, c, m)$ and $(W_1$, W_2, W_3, W_4, s'_1, s'_2, s'_3, s'_4, c', $m)$ where $s'_i \neq s_i$, $i = 1, 2, 3, 4$, $c' \neq c$, satisfying

$$
\begin{aligned}
&W_1^c a^{s_2 - c2^{l_e}} y^{s_3} = W_1^{c'} a^{s'_2 - c'2^{l_e}} y^{s'_3} \bmod n^2,\\
&g^{s_3} W_2^c = g^{s'_3} W_2^{c'} \bmod n^2,\\
&a^{s_4} a_0^{s_3} W_3^{-(s_1 - c2^{l_e})} = a^{s'_4} a_0^{s'_3} W_3^{-(s'_1 - c'2^{l_e})} \bmod n^2,\\
&g^{s_4} W_2^{-s_2 + c2^{l_z}} = g^{s'_4} W_2^{-s'_2 + c'2^{l_z}} \bmod n^2,\\
&g^{s_1 - c2^{l_e}} h^{s_3} W_4^c = g^{s'_1 - c'2^{l_e}} h^{s'_3} W_4^{c'} \bmod n^2.
\end{aligned}
$$

Let $\Delta s_i = s_i - s_i'$, $i = 1,2,3,4$, $\Delta c = c' - c$, then

$$a^{\Delta s_2 + \Delta c 2^{l_z}} y^{\Delta s_3} = W_1^{\Delta c} \bmod n^2, \tag{1}$$
$$g^{\Delta s_3} = W_2^{\Delta c} \bmod n^2, \tag{2}$$
$$a^{\Delta s_4} a_0^{\Delta s_3} = W_3^{\Delta s_1 + \Delta c 2^{l_e}} \bmod n^2, \tag{3}$$
$$g^{\Delta s_4} = W_2^{\Delta s_2 + \Delta c 2^{l_z}} \bmod n^2, \tag{4}$$
$$g^{\Delta s_1 + \Delta c 2^{l_e}} h^{\Delta s_3} = W_4^{\Delta c} \bmod n^2, \tag{5}$$

From formulas (1), (2), (5),we deduce

$$\Delta c | \Delta s_3, \Delta c | \Delta s_2, \Delta c | \Delta s_1, \tag{6}$$

otherwise Strong RSA assumption would be broken [32].

Similarly, from formula (4)

$$\Delta s_2 - \Delta c 2^{l_z} | \Delta s_4. \tag{7}$$

From formulas (2), (4),we deduce

$$\frac{\Delta s_3}{\Delta c} = \frac{\Delta s_4}{\Delta s_2 + \Delta c + 2^{l_z}}. \tag{8}$$

From formulas (6),(7),(8), we deduce $\Delta s_4 = \Delta c \frac{\Delta s_3}{\Delta c}(\frac{\Delta s_2}{\Delta c} + 2^{l_z})$, let $r = \frac{\Delta s_3}{\Delta c}$, $z_i = \frac{\Delta s_2}{\Delta c} + 2^{l_z}$, $e_i = \frac{\Delta s_1}{\Delta c} + 2^{l_e}$, it follows that $\frac{\Delta s_4}{\Delta c} = r z_i$.

From formula (3) and above results,

$$W_3^{e_i} = (a^{z_i} a_0)^r \bmod n^2,$$

which means W_3 has the form of $((a^{z_i} a_0)^{\frac{1}{e_i}})^r \bmod n^2$.

From checking the lengths of $s_1, s_2, s_1', s_2', c, c'$, we are assured that e_i, z_i are in the specified intervals with great probability.

Honest Verifier Zero-knowledge: For $W_1, W_2, W_3, W_4 \in Z_{n^2}^*$, select the following random values

$$s_1 \in \pm\{0,1\}^{\epsilon(k+\mu_e)+1}, s_2 \in \pm\{0,1\}^{\epsilon(k+\mu_z)+1}, s_3 \in \pm\{0,1\}^{\epsilon(k+2l_n-2)+1},$$
$$s_4 \in \pm\{0,1\}^{\epsilon(k+2l_n+l_z-2)+1}, c \in \{0,1\}^k,$$

then compute

$$R_1 = W_1^c a^{s_2 - c2^{l_e}} y^{s_3} \bmod n^2, \quad R_2 = g^{s_3} W_2^c \bmod n^2, \quad R_3 = a^{s_4} a_0^{s_3} W_3^{-(s_1 - c2^{l_e})} \bmod n^2,$$
$$R_4 = g^{s_4} W_2^{-s_2 + c2^{l_z}} \bmod n^2, \quad R_5 = g^{s_1 - c2^{l_e}} h^{s_3} W_4^c \bmod n^2).$$

The distribution of $(W_1, W_2, W_3, W_4, s_1, s_2, s_3, s_4, c, R_1, R_2, R_3, R_4, R_5)$ is statistically indistinguishable with that from the real prover. □

B Proof of Theorem 3.2

B.1 Preliminaries

Lemma B.1 (Generalized Forking Lemma [21]). *Consider a PPT (probabilistic polynomial time) algorithm $\mathcal{P}$, a PPT predicate Q, and a hash function $\mathcal{H}$ with range $\{0,1\}^k$ thought of as a random oracle. The predicate Q satisfies that $Q(x) = \top \Rightarrow \{x = (\rho_1, c, \rho_2) \ \wedge \ c = \mathcal{H}(\rho_1)\}$. $\mathcal{P}$ is allowed to ask queries on $\mathcal{H}$ and $\mathcal{R}$, where $\mathcal{R}$ is a process that given (t, c) reprograms $\mathcal{H}$ so that $\mathcal{H}(t) = c$, and it is assumed that $\mathcal{P}$ behaves in such a way that queries (t, c) to $\mathcal{R}$ adhere to the following conditions:*

1. *c is uniformly distributed over $\{0,1\}^k$.*
2. *The probability of the occurrence of a specific $t = t_0$ is upper bounded by $2/2^k$.*

Suppose that $\mathcal{P}^{\mathcal{H},\mathcal{R}}(\mathsf{param})$ returns a x such that $Q(x) = \top$ with non-negligible probability $\epsilon \geq 10(q_R+1)(q_R+q_H)/2^k$, where q_R, q_H are numbers of queries to $\mathcal{R}$ and $\mathcal{H}$ respectively. Then there exists a PPT $\mathcal{P}'$ so that if $y \leftarrow \mathcal{P}'(\mathsf{param})$ it holds with probability 1/9 that (1) $y = (\rho_1, c, \rho_2, c', \rho_2)$, (2) $Q(\rho_1, c, \rho_2) = \top$ and $Q(\rho_1, c', \rho_2') = \top$, (3) $c \neq c'$. The probabilities are taken over the choices for $\mathcal{H}$, the random coin tosses of $\mathcal{P}$ and the random choice of the public parameters param.

B.2 Proof of Theorem 3.2

Proof. Suppose there exists an adversary $\mathcal{A}$ breaking traceability of the proposal, i.e., $\mathcal{A}$ is able to non-negligibly output a valid group signature $(m, W_1, W_2, W_3, W_4, s_1, s_2, s_3, s_4, c)$ which can be opened to a group member ID_{i^*} who has not been queried to $\mathcal{O}_{a-join}$, and group signature of m by this member has not been queried to $\mathcal{O}_{sign}$. An algorithm $\mathcal{B}$ can be constructed to resolve Strong RSA problem in QR_{n^2}, i.e., calculating $(u, e > 1)$ that $u^e = z \bmod n^2$ given a random $z \in QR_{n^2}$, utilizing $\mathcal{A}$.

$\mathcal{B}$ selects $x \in_R Z^*_{n^2}$, chooses $g, h \in_R Z^*_{n^2}$ which are quadratic residues module n^2 with high probability. $\mathcal{B}$ sets $y = g^x \bmod n^2$, $a = z^{\prod_{i\in[1,Q]} e_i}$, $a_0 = a^r$, where $r \in Z^*_{n^2}$ and $e_i, i \in [1, Q]$ are randomly chosen primes from Γ. Q is the maximum query number to O_{a-join}.

$\mathcal{B}$ simulates answers to the following queries.

$\mathcal{O}_{pub}$: returns $gpk = (n, a_0, a, g, h, y, l, \mu, l_e, \mu_e, l_z, \mu_z, \epsilon)$.

$\mathcal{O}_{a-join}$: $\mathcal{A}$ will sends C_i and a proof of knowledge of $x_i \in S(2^{l_x}, 2^{\mu_x})$ that $C_i a^{-ID_i} = a^{nx_i}$. $\mathcal{B}$ rewinds $\mathcal{A}$ and provides a new random challenge, then extracts x_i from the two proofs of knowledge obtained from $\mathcal{A}$ (the detail of the rewind technique is referred to Section 6 of [33]). $\mathcal{B}$ then returns (e_i, A_i), where $A_i = z^{(ID_i+nx_i+r)\prod_{j\neq i} e_j}$. $\mathcal{B}$ maintains a list L of (ID_i, x_i, e_i, C_i).

$\mathcal{O}_{open}(\sigma)$: Given a group signature $\sigma = (W_1, W_2, W_3, W_4, s_1, s_2, s_3, s_4, c)$ of m, $\mathcal{B}$ declines the query if σ is not valid, otherwise decrypts (W_1, W_2) to get $C = W_1/W_2^x$ since it knows the decryption key x. If there exists a C_j in L that $C_j = C$, then $\mathcal{B}$ generates a proof of knowledge $\tau_1 = SK_1\{x : y = g^x \bmod n^2, W_2^x = W_1/C \bmod n^2\}$, and simulates another proof of knowledge $\tau_2 = SK_2\{\gamma : (a^{ID_j}/C)^\gamma = 1 \bmod n^2, (a^n)^\gamma = 1 \bmod n^2\}$. $\mathcal{B}$ returns $(ID_j, C, \tau_1, \tau_2)$. If C does not exist in L, $\mathcal{B}$ just generates τ_1 and returns $(NULL, C, \tau_1)$.

Now $\mathcal{A}$ outputs a valid group signature $(W_1, W_2, W_3, W_4, s_1, s_2, s_3, s_4, c)$ of m by $ID_{i^*}, i^* \notin [1, Q]$. Apply Lemma B.1 to $\mathcal{A}$, where VERIFY is the predicate, $\mathcal{B}$ will get a $\mathcal{A}'$ that outputs $(m, W_1, W_2, W_3, W_4, s_1, s_2, s_3, s_4, c)$ and $(m, W_1, W_2, W_3, W_4, s'_1, s'_2, s'_3, s'_4, c' \neq c)$, then $\mathcal{B}$ can extract (r^*, e^*, z^*) satisfying ${W_3}^{e^*} = (a^{z^*} a_0)^{r^*} \bmod n^2$, and $z^* \neq ID_i \bmod n$ for any $i \in [1, Q]$ queried by $\mathcal{A}$.

If $(W_3, e^*, z^*) = (A_I^{r^*}, e_I, nx_I + ID_I)$ for some $I \in [1, Q]$, let $z^* = ID_{i^*} + nx^*$, then $a^{ID_I - ID_{i^*} + n(x_I - x^*)} = 1 \bmod n^2$, so $np'q' | (ID_I - ID_{i^*} + n(x_I - x^*))$, from $|ID_I - ID_{i^*}| < n$, we can get $ID_I = ID_{i^*}$; then $a^{n(x_I - x^*)} = 1 \bmod n^2$, since $ord(a^n) = p'q'$, we can get $p'q' | (x_I - x^*)$, so $x_I = x^*$ from $|x_I - x^*| < p'q'$. Then $ID_{i^*} = ID_I$, i.e., ID_{i^*} has been queried to $\mathcal{O}_{a-join}$, a contradiction to the presumption for $\mathcal{A}$.

So it must be $(W_3, e^*, z^*) \neq (A_i^{r^*}, e_i, nx_i + ID_i)$ for any $i \in [1, Q]$, then (W_3, e^*, z^*) is a breaking of coalition resistance of ACJT scheme (Section 2.3) under public key $(a^{r^*}, a_0^{r^*})$ (Lemma 2.1). Although Lemma 2.1 is in QR_n, it is evident that it can also be proved in QR_{n^2} following the proof in [6]. The Strong RSA problem can be resolved quite similarly to the proof of Lemma 2.1. □

C Proof of Theorem 3.3

Proof. Suppose there exists an adversary $\mathcal{A}$ breaking anonymity of the proposal, i.e., $\mathcal{A}$ can output $b' = b$ with non-negligible probability given a group signature of m by ID_{i_b}, where $b \in \{0, 1\}$ is chosen by $\mathcal{O}_{ch}$. Then we can construct an algorithm $\mathcal{B}$ resolving DDH problem in QR_{n^2} when $\mathcal{B}$ does not know the factor of n.

Given $(g^\alpha, g^\beta, g^\gamma)$, where $\alpha, \beta \in Z_{n^2}^*$, $\mathcal{B}$ is to decide if $\gamma = \alpha\beta$, or just an random value independent from α, β.

$\mathcal{B}$ selects $x \in_R Z_{n^2}^*$, sets $y = g^x \bmod n^2$, $h = g^\beta$, $a = g^{r_1 \prod_{i \in [1,Q]} e_i}$, $a_0 = a^{r_1}$, where $r_1, r_2 \in_R Z_{n^2}^*$ and $e_i, i \in [1, Q]$ are randomly chosen primes in Γ. Q is the maximum query number to O_{a-join}.

Game G_1:

$\mathcal{B}$ simulates the following queries.

$\mathcal{O}_{pub}$: returns $gpk = (n, a_0, a, g, h, y, l, \mu, l_e, \mu_e, l_z, \mu_z, \epsilon)$.

$\mathcal{O}_{a-join}$: $\mathcal{A}$ will sends C_i and a proof of knowledge of $x_i \in S(2^{l_x}, 2^{\mu_x})$

that $C_i a^{-ID_i} = a^{nx_i}$. $\mathcal{B}$ rewinds $\mathcal{A}$ and provides a new random challenge, then extracts x_i from the two proofs of knowledge. The detail of the rewind technique is referred to Section 6 of [33]. $\mathcal{B}$ then returns (e_i, A_i), where $A_i = g^{r_1(ID_i+nx_i+r_2)\prod_{j\neq i} e_j}$. $\mathcal{B}$ maintains a list L of (ID_i, x_i, e_i, C_i). ID_i is marked U^a.

$\mathcal{O}_{open}(\sigma)$: Given a group signature $\sigma = (W_1, W_2, W_3, W_4, s_1, s_2, s_3, s_4, c)$ of m, $\mathcal{B}$ declines the query if $\sigma = \sigma_b$ or σ is not valid, otherwise decrypts (W_1, W_2) to get $C = W_1/W_2^x$ since it knows the decryption key x, if there exists a C_j in L that $C_j = C$, then $\mathcal{B}$ generates a proof of knowledge $\tau_1 = SK_1\{x : y = g^x \bmod n^2, W_2^x = W_1/C \bmod n^2\}$, and simulates another proof of knowledge $\tau_2 = SK_2\{\eta : (a^{ID_j}/C)^\eta = 1 \bmod n^2, (a^n)^\eta = 1 \bmod n^2\}$. $\mathcal{B}$ returns $(ID_j, C, \tau_1, \tau_2)$. If C does not exist in L, $\mathcal{B}$ just generates τ_1 and returns $(NULL, C, \tau_1)$.

$\mathcal{O}_{ch}(b, ID_{i_0}, ID_{i_1}, m)$: If ID_{i_0}, ID_{i_1} are marked U^a, $\mathcal{B}$ retrieves $(ID_{i_k}$, x_{i_k}, e_{i_k}, $C_{i_k})$, $k = \{0,1\}$ from list L, sets $W_1 = a^{ID_{i_b}+nx_{i_b}}(g^\alpha)^x, W_2 = g^\alpha, W_3 = A_i^\alpha = (g^\alpha)^{r_1(ID_{i_b}+nx_{i_b}+r_2)\prod_{j\neq i_b} e_j}, W_4 = g^{e_{i_b}}(g^\gamma)$, then simulates τ, a proof of knowledge of $(e_{i_b}, z_{i_b}, \alpha, z_{i_b}\alpha)$, where $z_{i_b} = ID_{i_b} + nx_{i_b}$, as in the proof of honest verifier zero-knowledge (Appendix A). $\mathcal{B}$ returns $\sigma_b = (W_1, W_2, W_3, W_4, \tau)$.

$\mathcal{B}$ outputs 1 if $b' = b$ (implying $\gamma = \alpha\beta$), outputs 0 otherwise (implying γ random).

If $\gamma = \alpha\beta$, then σ_b is a perfect group signature of m by ID_{i_b}, which is more advantageous for $\mathcal{A}$ to win than the case of random γ.

Game G_2:

$\mathcal{B}$ simulates oracle queries similarly to **Game** G_1 except $\mathcal{O}_{ch}(b$, ID_{i_0}, ID_{i_1}, $m)$.

$\mathcal{O}_{ch}(b, ID_{i_0}, ID_{i_1}, m)$: If ID_{i_0}, ID_{i_1} are marked U^a, $\mathcal{B}$ retrieves $(ID_{i_k}$, x_{i_k}, e_{i_k}, $C_{i_k})$, $k = \{0,1\}$ from list L, sets $W_1 = a^{ID_{i_b}+nx_{i_b}}(g^\alpha)^x, W_2 = g^\alpha$, $\underline{W_3 = A_i^{r'}}$, $(r' \in_R Z_{n^2}^*), W_4 = g^{e_{i_b}}(g^\gamma)$, then simulates τ, a proof of knowledge of $(e_{i_b}, z_{i_b}, \alpha, z_{i_b}\alpha)$, where $z_{i_b} = ID_{i_b} + nx_{i_b}$. $\mathcal{B}$ returns $\sigma_b = (W_1, W_2, W_3, W_4, \tau)$.

The difference between G_1 and G_2 is that (g, A_{i_b}, W_2, W_3) is a DDH quadruple in G_1, while a random quadruple in G_2.

Game G_3:

$\mathcal{B}$ simulates oracle queries similarly to **Game** G_1 except $\mathcal{O}_{ch}(b$, ID_{i_0}, ID_{i_1}, $m)$.

$\mathcal{O}_{ch}(b, ID_{i_0}, ID_{i_1}, m)$: If ID_{i_0}, ID_{i_1} are marked U^a, $\mathcal{B}$ retrieves $(ID_{i_k}$, x_{i_k}, e_{i_k}, $C_{i_k})$, $k = \{0,1\}$ from list L, sets $W_1 = a^{ID_{i_b}+nx_{i_b}}(g^\alpha)^x$, $\underline{W_2 = g^{r''}}$, $W_3 = A_i^{r'}$, $(r', r'' \in_R Z_{n^2}^*), W_4 = g^{e_{i_b}}(g^\gamma)$, then simulates τ, a proof of knowledge of $(e_{i_b}, z_{i_b}, \alpha, z_{i_b}\alpha)$, where $z_{i_b} = ID_{i_b} + nx_{i_b}$. $\mathcal{B}$ returns $\sigma_b = (W_1, W_2, W_3, W_4, \tau)$.

The difference between G_2 and G_3 is that $(y, g, W_1/a^{z_{i_b}}, W_2)$ is a DDH quadruple in G_2, while a random quadruple in G_3.

Denote $\mathcal{A}$'s output in Game G_i as $\mathcal{A}^{G_i}$, then suppose $\mathcal{A}$ is a successful adversary against anonymity attacks, that is $\exists\epsilon > 0$ which is non-negligible, so that

$$|P[\mathcal{A}^{G_1} = b|\gamma = \alpha\beta] - 1/2| \geq \epsilon.$$

It is easy to see that for $b \in \{0, 1\}$,

$$|P[\mathcal{A}^{G_1} = b|\alpha, \beta, \gamma \in_R Z^*_{n^2}] - P[\mathcal{A}^{G_2} = b|\alpha, \beta, \gamma \in_R Z^*_{n^2}]| \leq Adv^{DDH}_{\mathcal{A}} \leq Adv^{DDH},$$

$$|P[\mathcal{A}^{G_2} = b|\alpha, \beta, \gamma \in_R Z^*_{n^2}] - P[\mathcal{A}^{G_3} = b|\alpha, \beta, \gamma \in_R Z^*_{n^2}]| \leq Adv^{DDH}_{\mathcal{A}} \leq Adv^{DDH},$$

where Adv^{DDH} is the maximum value for all algorithm $\mathcal{A}$.

In Game G_3, every component of the challenge is randomized independently if $\alpha, \beta, \gamma \in_R Z^*_{n^2}$, so there exists a negligible ϵ_1

$$|P[\mathcal{A}^{G_3} = b|\alpha, \beta, \gamma \in_R Z^*_{n^2}] - 1/2| < \epsilon_1,$$

So

$$\begin{aligned}
\epsilon &\leq |P[\mathcal{A}^{G_1} = b|\gamma = \alpha\beta] - 1/2| \\
&= |P[\mathcal{A}^{G_1} = b|\gamma = \alpha\beta] - P[\mathcal{A}^{G_1} = b|\alpha, \beta, \gamma \in_R Z^*_{n^2}]| \\
&\quad + |P[\mathcal{A}^{G_1} = b|\alpha, \beta, \gamma \in_R Z^*_{n^2}] - P[\mathcal{A}^{G_2} = b|\alpha, \beta, \gamma \in_R Z^*_{n^2}]| \\
&\quad + |P[\mathcal{A}^{G_2} = b|\alpha, \beta, \gamma \in_R Z^*_{n^2}] - P[\mathcal{A}^{G_3} = b|\alpha, \beta, \gamma \in_R Z^*_{n^2}]| \\
&\quad + |P[\mathcal{A}^{G_3} = b|\alpha, \beta, \gamma \in_R Z^*_{n^2}] - 1/2| \\
&\leq |P[\mathcal{B}^{G_1} = 1|\gamma = \alpha\beta] - P[\mathcal{B}^{G_1} = 1|\alpha, \beta, \gamma \in_R Z^*_{n^2}]| + 2Adv^{DDH} + \epsilon_1 \\
&\leq Adv^{DDH}_{\mathcal{B}} + 2Adv^{DDH} + \epsilon_1 \\
&\leq 3Adv^{DDH} + \epsilon_1
\end{aligned}$$

This is a contradiction to DDH assumption. □

D Proof of Theorem 3.4

Proof. Suppose there exists an adversary $\mathcal{A}$ breaking the non-frameability of the proposal, i.e., $\mathcal{A}$ is able to successfully produce a valid group signature that can be opened to an honest group member ID_I who is not queried to $\mathcal{O}_{b-join}$ by $\mathcal{A}$, and $\mathcal{O}_{sign}(ID_I, m)$ is not queried. We can utilize $\mathcal{A}$ to construct an adversary $\mathcal{B}$ who is given factors of n, breaking Discrete Logarithm Assumption in QR_{n^2}, i.e., given a random DLA instance $b \in_R QR_{n^2}$, to calculate δ that $a^\delta = b \bmod n^2$.

$\mathcal{B}$ selects group secret key and public key exactly as GM does in the proposal since it knows the factor of n. Additionally $\mathcal{B}$ sets a random variable I with value randomly chosen from $\{1, ..., Q\}$, where Q is the maximum query number made by $\mathcal{A}$.

$\mathcal{B}$ simulates answers to the following queries.

$\mathcal{O}_{pub}$: returns $gpk = (n, a_0, a, g, h, y, l, \mu, l_e, \mu_e, l_z, \mu_z, \epsilon)$.

$\mathcal{O}_{key}$: returns $(x, p, q, p'q')$.

$\mathcal{O}_{b-join}$: $\mathcal{B}$ selects an ID_i, computes $C_i = a^{ID_i} a^{nx_i}$ and π_i, where $x_i \in_R S(2^{l_x}, 2^{\mu_x})$, π_i is a proof of knowledge of such a x_i. If $i = I$, $\mathcal{B}$ sets $x_I = \log_a b$ which is unknown to itself, computes $C_I = a^{ID_I} b^n \bmod n^2$, simulates a proof of knowledge of x_I. $\mathcal{B}$ returns (ID_i, C_i, π_i) and waits for a response from $\mathcal{A}$. $\mathcal{A}$ should return (e_i, A_i), where $A_i^{e_i} = C_i a_0 \bmod n^2$ or a signal of failure. $\mathcal{B}$ maintains a list L of (ID_i, x_i, e_i, C_i). ID_i is marked U^b.

$\mathcal{O}_{sign}(ID_j, m)$: If ID_j is marked U^b and $j \neq I$, $\mathcal{B}$ knows the member secret key and member certificate, so $\mathcal{B}$ just generates a group signature exactly as SIGN. If $j = I$, $\mathcal{B}$ does not know the member secret key of ID_I, but it will simulate a group signature as in proof of honest verifier zero-knowledge (Appendix A).

If $\mathcal{A}$ outputs a group signature $\sigma = (m, W_1, W_2, W_3, W_4, s_1, s_2, s_3, s_4, c)$ that is opened to ID_I with non-negligible probability, $\mathcal{B}$ can derive an algorithm $\mathcal{A}'$ which can output another group signature $(m, W_1, W_2, W_3, W_4, s_1', s_2', s_3', s_4', c' \neq c)$ which is opened to ID_I too according to Lemma B.1. Then $\mathcal{B}$ can extract (r^*, e^*, z^*) that $W_3^{e^*} = (a^{z^*} a_0)^{r^*} \bmod n^2$. Because σ is opened to member ID_I, then there must exist a x^* that $ID_I + nx^* = z^*$. We show that $x^* = x_I$.

If not, i.e., $nx_I + ID_I \neq nx^* + ID_I$. But we have $a^{ID_I + nx_I} = a^{ID_I + nx^*} \bmod n^2$ from OPEN and JUDGE, i.e., $a^{n(x_I - x^*)} = 1 \bmod n^2$, that is $p'q' | (x_I - x^*)$ since $ord(a^n) = p'q'$. Because $|x_I - x^*| < p'q'$, it follows that $x^* = x_I = \log_a b$.

Thus PDL assumption in QR_{n^2} is broken. □