Group Signature where Group Manager, Members and Open Authority are Identity-Based

Victor K. Wei[1], Tsz Hon Yuen[1] and Fangguo Zhang[2]

[1] Department of Information Engineering,
The Chinese University of Hong Kong,
Shatin, Hong Kong
{kwwei, thyuen4}@ie.cuhk.edu.hk
[2] Department of Electronics and Communication Engineering,
Sun Yat-sen University,
Guangzhou, 510275, P.R.China
isdzhfg@zsu.edu.cn

Abstract. We present the first group signature scheme with provable security and signature size $O(\lambda)$ bits where the group manager, the group members, and the Open Authority (OA) are all identity-based. We use the security model of Bellare, Shi, and Zhang [3], except to add three identity managers for manager, members, and OA respectively, and we discard the Open Oracle ($\mathcal{OO}$). Our construction uses identity-based signatures summarized in Bellare, Namprempre, and Neven [2] for manager, Boneh and Franklin's IBE [7] for OA, and we extend Bellare et al.[3]'s group signature construction by verifiably encrypt an image of the member public key, instead of the public key itself. The last innovation is crucial in our efficiency; otherwise, Camenisch and Damgard[9]'s verifiable encryption would have to be used resulting in lower efficiency.

1 Introduction

Identity based cryptography, introduced by Shamir [25], allows the users' public key to be their identity. Usually a trusted third party computes the private key from the identity (any arbitrary string such as email address). Comparing with certificate from certificate authority (CA), the identity based public key can identify the user immediately. Besides, the problem of distribution of public keys is avoided in identity based cryptography.

Group signature, introduced by Chaum and van Heyst [13], allows any member of a group to sign on behalf of the group. However, the identity of the signer is kept secret. Anyone can verify that the signature is signed by a group member, but cannot tell which one. Therefore group signature provides anonymity for signers. Usually in group signature schemes, a group manager issues certificates to his group members. Then the group member uses his certificate and his own secret key to sign messages. Anyone can verify the signature by the group manager's public key. In some cases, an open authority has a secret key to revoke the anonymity of any signature in case of dispute. Mostly it can be done by an encryption to the open authority when signing the message. On the other hand, anonymity can be revoked when a signer double signs in some schemes. Group signature is a very useful tool in real world. It can be used in e-cash, e-voting or attestation [8] in trusted computing group.

Weil and Tate pairing has been widely used in identity based cryptography in recent years. Pairing is also used to construct short group signature [6] recently. However, none of the existing group signature scheme can be completely verified in an identity based manner, that is the group public key and the opener public key are arbitrary strings. The current "Identity based" group signature are mostly for identity based group member only ([22][19][26][11][14]). We think that identity based group member is not enough for group signature. It is because the signer's public key is always anonymous in group signature. Whether it is identity based or not has no effect to the verifier. We think that it is constructive to have a group signature with identity based group public key, which is the identity of the group manager in this case. At the same time, we also want to support identity based group members, as well as open authority. We call this new scheme to be a fully identity based group signature. In this paper, we will give a generic construction, and then a specific instantiation of such a identity based group signature.

Contributions. Our main contributions are:

- We introduce the formal study of group signature schemes with identity based group manager, identity based group members and identity based open authority.
- We present the first construction of the above scheme, complete with security models, and reductionist security proofs in the random oracle model. The size of the signature is $O(\lambda)$ bits.
- We extend Bellare, Shi, and Zhang [3]'s generic group signature construction by verifiably encrypt, to the Open Authority (OA), a one-way image of the signer public key instead of the signer public key itself. This technique is crucial to the topic in this paper.

The rest of the paper is **organized** as follows: Section 2 contains preliminaries. Section 3 contains the security model. Section 4 contains the constructions. Section 5, security theorems. Section 6, discussions and applications.

2 Preliminaries

2.1 Related results

After the introduction of group signature by Chaum and van Heyst [13], there are numerous group signature schemes proposed, such as Ateniese et al [1], Dodis et al [15], Boneh et al [6]. The state-of-the-art is to have a group signature scheme with signature size independent of the group size. The security model of dynamic group signature is proposed in [3].

Identity based signature is suggested in 1984 by Shamir [25], but practical identity based encryption is not found until 2001 by Boneh and Franklin [7] using Weil pairing. Identity based group signature is firstly proposed by Park et al [22]. [19] showed that the anonymity of the scheme was not guaranteed. Tseng and Jan [26] presented a novel ID-based group scheme. However, it is universally forgeable [18] and not coalition-resistant [17]. Several identity based group signature schemes are proposed in [11], [14]. [11] requires a new pair of certificate for each signature. However all of them only have identity based key pairs for group members only. Group signature scheme with identity based group manager and identity based open authority remains as an open problem.

2.2 Pairings

Following the notation of pairings in [7], let $\mathbb{G}_1, \mathbb{G}_2$ be (mutiplicative) cyclic groups of prime order p. Let g_1 be a generator of $\mathbb{G}_1$ and g_2 be a generator of $\mathbb{G}_2$. Let ψ is a computable isomorphism from $\mathbb{G}_1$ to $\mathbb{G}_2$, with $\psi(g_2) = g_1$.

Definition 1. *A map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is called a bilinear pairing if, for all $x \in \mathbb{G}_1, y \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p$, we have $\hat{e}(x^a, y^b) = \hat{e}(x, y)^{ab}$, and $\hat{e}(g_1, g_2) \neq 1$.*

Definition 2. *(co-CDH problem) The co-computational Diffie-Hellman problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is as follows: given $P, P^\alpha \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, for unknown $\alpha \in \mathbb{Z}_p$, to compute Q^α.*

Definition 3. *(DDH problem) The decisional Diffie-Hellman problem in $\mathbb{G}_1$ is as follows: given $P, P^\alpha, P^\beta, R \in \mathbb{G}_1$ for unknown $\alpha, \beta \in \mathbb{Z}_p$, to decide if $R = P^{\alpha\beta}$.*

Definition 4. *(co-DBDH problem) The co-decisional Bilinear Diffie-Hellman problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is as follows: given $P, P^\alpha, P^\beta \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, $R \in \mathbb{G}_T$ for unknown $\alpha, \beta \in \mathbb{Z}_p$, to decide if $R = \hat{e}(P, Q)^{\alpha\beta}$.*

Definition 5. *(k-SDH' problem) The k-Strong Diffie-Hellman' problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is as follows: given $g_1, {g_1}^\gamma, ..., {g_1}^{\gamma^k} \in \mathbb{G}_1$ and $g_2, {g_2}^\gamma \in \mathbb{G}_2$ as input, outputs a pair $({g_1}^{1/\gamma+x}, x)$ where $x \in \mathbb{Z}_p^*$.*

Definition 6. *(k-CAA2 problem) The k-CAA2 problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is as follows: given $v, u \in \mathbb{G}_1$, $g_2, {g_2}^\gamma \in \mathbb{G}_2$ and pairs (A_i, e_i, λ_i) with distinct and nonzero e_i's satisfying $A_i^{\gamma+e_i} v^{\lambda_i} = u$ for $1 \leq i \leq k$ as input, outputs a pair $(A_{k+1}, e_{k+1}, \lambda_{k+1})$ satisfying $A_{k+1}^{\gamma+e_{k+1}} v^{\lambda_{k+1}} = u$, with $e_{k+1} \neq e_i$ for all $1 \leq i \leq k$.*

The above k-SDH' problem and k-CAA2 problem are proven equivalent in [27] assume the value $log_u(v)$ is known. [27] also shows that the k-Strong Diffie-Hellman assumption in [20],[5],[28] is at least as strong as the k-SDH' problem.

Definition 7. *Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be a pairing. Given the following:*

1. $g_1, g_1^\alpha, g_1^{\beta_i}, g_1^{\gamma_i} \in \mathbb{G}_1$ *for* $1 \leq i \leq k$;
2. $g_2, g_2^{\delta_1}, g_2^{\delta_2} \in \mathbb{G}_2$, $R \in G_T$;

3. $\Pr\{\gamma_i = \alpha\beta_i,\ all\ i, 1 \le i \le k\} = \Pr\{\gamma_i \ne \alpha\beta_i,\ all\ i, 1 \le i \le k\} = 1/2.$
4. $\Pr\{\gamma_i = \alpha\beta_i,\ all\ i, 1 \le i \le k\ AND\ R = \hat{e}(g_1, g_2)^{\delta_1\delta_2}\} = \Pr\{\gamma_i \ne \alpha\beta_i,\ all\ i, 1 \le i \le k\ AND\ R \ne \hat{e}(g_1, g_2)^{\delta_1\delta_2}\} = 1/2$

The Lockstep DDH Problem *(resp.* Lockstep DDH+coDBDH Problem*) is to distinguish between the two nonzero probability events in (3) (resp. (4)) above with non-negligible probability over 1/2. The* Lockstep DDH Assumption *(resp.* Lockstep DDH+coDBDH Assumption*) is that no PPT algorithm can solve the Lockstep DDH Problem (resp. Lockstep DDH+coDBDH Problem).*

Lemma 1 *The Lockstep DDH Assumption in $\mathbb{G}_1$ holds if and only if the DDH Assumption in $\mathbb{G}_1$ holds. The Lockstep DDH+coDBDH Assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if and only if the DDH Assumption in $\mathbb{G}_1$ and the co-DBDH assumption in $(\mathbb{G}_2, \mathbb{G}_1)$ both hold.*

Proof. We prove for the Lockstep DDH+coDBDH Assumption only. The other case is similar. The DDH assumption and the co-DBDH assumption implies the Lockstep DDH+coDBDH assumption is straightforward.

We now proof in the opposite direction. Let $\mathcal{B}$ be a PPT solver of the Lockstep DDH+coDBDH problem with advantage ϵ_1. Consider its performance when given the following problems: [DDHi: $(g_1, g_1^{\alpha}, g_1^{\beta_i}, g_1^{\gamma_i})$] for $1 \le i \le k$ and [co-DBDH: $(g_1, g_2, g_2^{\delta_1}, g_2^{\delta_2}, R)$]; where $\gamma_i = \alpha\beta_i$ or is random with half-half probability, and $R = \hat{e}(g_1, g_2)^{\delta_1\delta_2}$ or is random with half-half probability. Then we can give the "generalized lockstep" problem to $\mathcal{B}$ to solve: $[(g_1, g_1^{\alpha}, g_1^{\beta_i}, g_1^{\gamma_i})$ for $1 \le i \le k$; $(g_2, g_2^{\delta_1}, g_2^{\delta_2}, R)]$. With probability $2^{-(k+1)}$, the "generalized lockstep" problem is a Lockstep DDH+coDBDH problem, and in that case $\mathcal{B}$ solves it with probability $1/2 + \epsilon_1$. Otherwise, the "generalized lockstep" problem is not a Lockstep DDH+coDBDH problem, and let us consider $\mathcal{B}$'s performance in this case. Let ϵ_2 denote the probability that $\mathcal{B}$ outputs $\perp$ meaning the problem is not a Lockstep DDH+coDBDH problem. $\epsilon_2 = 0$ if he is not allowed to do so. Then $\mathcal{B}$ outputs either DDHi and co-DBDH decision with equal probability $(1 - \epsilon_2)/2$ because there is a symmetry w.r.t. the two cases.

Let us build an algorithm $\mathcal{B}$' to solve DDHi and co-DBDH: $\mathcal{B}$' outputs "yes" to DDHi and co-DBDH if $\mathcal{B}$ outputs "yes" on input "generalized lockstep" problem; and $\mathcal{B}$' outputs "no" otherwise. Then:

$$\sum_{i=1}^{k} \tfrac{1}{k+1}\Pr\{\mathcal{B}\text{' solves DDH}i\} + \tfrac{1}{k+1}\Pr\{\mathcal{B}' \text{ solves co-DBDH}\}$$
$$= \tfrac{1}{2^{k+1}} \Pr\{\mathcal{B} \text{ solves Lockstep DDH+coDBDH}\} + \epsilon_2 + \tfrac{1}{2}(1 - \tfrac{1}{2^{k+1}} - \epsilon_2)$$

Therefore $\mathcal{B}$' has a probability non-negligibly over half of solving either DDHi or co-DBDH problem. □

3 Security Model

We present a security model for the identity based group signature. Here we adapt the models for dynamic group signature in [3], and add support for IBGS. Our scheme is applicable to multiple certificate authorities (CA, or group managers) and open authorities (OA).

3.1 Syntax

A *identity-based group signature (IBGS)* is a tuple (Init, OKg, GKg, UKg, Join, Iss, GSig, GVf, Open, Judge) where:

- Init: $1^\lambda \mapsto \mathsf{param}$. On input the security parameter 1^λ, generates system-wide public parameters param. The identity manager of CA (IM_A) has (sk,pk) pair (x_A, y_A) for CA (resp. IM_U has (x_U, y_U) for group members, IM_O has (x_O, y_O) for OA) and an efficiently samplable one-way NP-relation $\langle \mathcal{R}_A \rangle$, with trapdoor x_A (resp. $\langle \mathcal{R}_U \rangle$, with trapdoor x_U, $\langle \mathcal{R}_O \rangle$, with trapdoor x_O). An efficiently samplable family of one-way NP-relation $\mathcal{F} = \{\langle \mathcal{R}_{C,i} \rangle : i\}$ with trapdoor gsk_i, is defined for issuing certificate. param is (y_A, y_U, y_O, $\mathcal{R}_A$, $\mathcal{R}_U$, $\mathcal{R}_O$, $\mathcal{F}$).
- OKg:$(\mathsf{oa}, x_O) \mapsto (x_{\mathsf{oa}}, aux_{\mathsf{oa}})$. On input the OA identity oa, the IM_O uses his secret key x_O to compute the secret key x_{oa} of the OA, some auxiliary information aux_{oa} such that $((x_{\mathsf{oa}}, aux_{\mathsf{oa}}), \mathsf{oa}) \in \mathcal{R}_O$.
- GKg: $(\mathsf{ca}, x_A) \mapsto (x_{\mathsf{ca}}, aux_{\mathsf{ca}}, \langle \mathcal{R}_{C,\mathsf{ca}} \rangle)$. On input the CA identity ca, the IM_A samples $\mathcal{F}$ to get the relation $\langle \mathcal{R}_{C,\mathsf{ca}} \rangle$. The IM_A uses his secret key x_A to compute the group secret key of CA x_{ca}, some auxiliary information aux_{ca} such that $((x_{\mathsf{ca}}, aux_{\mathsf{ca}}), \mathsf{ca}) \in \mathcal{R}_A$.
- UKg: $(\mathsf{id}, x_U) \mapsto (x_{\mathsf{id}}, aux_{\mathsf{id}})$. On input the identity id of the member, the IM_U uses his secret key x_U to compute the secret key x_{id} of the member, some auxiliary information aux_{id} such that $((x_{\mathsf{id}}, aux_{\mathsf{id}}), \mathsf{id}) \in \mathcal{R}_U$.
- Join,Iss is a pair of interactive protocols between the user and the CA, with common inputs ca and id. Iss's additional inputs are x_{ca} and aux_{ca}. Join's additional inputs are x_{id} and aux_{id}. At the conclusion, Join obtains $cert_{\mathsf{id}}$ satisfying $((x_{\mathsf{id}}, aux_{\mathsf{id}}, cert_{\mathsf{id,ca}}), \mathsf{id}) \in \mathcal{R}_{C,\mathsf{ca}}$, and Iss stores $(\mathsf{id}, cert_{\mathsf{id,ca}})$ in a registration table reg.
- GSig: (id, x_{id}, aux_{id}, ca, oa, $cert_{\mathsf{id,ca}}$, M) $\mapsto \sigma$. On input the keys, certificates and message, outputs a signature σ.
- GVf: $(\mathsf{ca}, \mathsf{oa}, M, \sigma) \mapsto 0$ or 1. On input the message and signature, outputs 1 for valid signature and 0 for invalid signature.
- Open: $(\mathsf{ca}, x_{\mathsf{oa}}, \mathsf{reg}, M, \sigma) \mapsto (i, \omega)$. The OA with key x_{oa} has read access to reg. On input a valid signature σ for message M for ca, output identity i for the corresponding signer, and ω is the proof of this claim. Output $i = \perp$ if no such member is found.
- Judge: $(\mathsf{ca}, \mathsf{id}, \mathsf{oa}, M, \sigma, \omega) \mapsto 0$ or 1. It checks if the proof ω is a valid proof that id is the real signer of σ for message M under ca, oa. Outputs 1 for valid and 0 for invalid.

Remarks: Here we use (param, ca) to denote *gpk* in [3]'s original syntax. We also split the GKg in [3] into Init, OKg and GKg. It is because we want to emphasize that group managers (CA) and open authorities (OA) are identity based.

3.2 Security notions

We have the security notions of Correctness, Anonymity, Traceability, Non-frameability from [3], with modification for identity based. We give a brief description here.

Correctness: Let $\sigma \leftarrow \mathsf{GSig}(\mathsf{id},\ x_{\mathsf{id}},\ aux_{\mathsf{id}},\ \mathsf{ca},\ \mathsf{oa},\ cert_{\mathsf{id},\mathsf{ca}},\ M)$ for arbitrary id, x_{id}, aux_{id}, ca, oa, $cert_{\mathsf{id},\mathsf{ca}}$, M. The IBGS has opening correctness if $(\mathsf{id}, \omega) \leftarrow \mathsf{Open}(\mathsf{ca}, x_{\mathsf{oa}}, \mathsf{reg}, M, \sigma)$ and $\mathsf{Judge}(\mathsf{ca}, \mathsf{id}, \mathsf{oa}, M, \sigma, \omega) = 1$ with overwhelming probability. It has verification correctness if $\mathsf{GVf}(\mathsf{ca}, \mathsf{oa}, M, \sigma) = 1$ with probability 1. The IBGS is correct if it has verification and opening correctness.

We have the following oracles for the adversary to query:

- The *Random Oracle* $\mathcal{RO}$: simulate the random oracle normally.
- The *Key Extraction Oracle-CA* $\mathcal{KEO}_c$: $\mathsf{ca} \to x_{\mathsf{ca}}$. Upon input CA ca, outputs his secret key x_{ca}.
- The *Key Extraction Oracle-OA* $\mathcal{KEO}_o$: $\mathsf{oa} \to x_{\mathsf{oa}}$. Upon input OA oa, outputs his secret key x_{oa}.
- The *Key Extraction Oracle-User* $\mathcal{KEO}_u$: $\mathsf{id} \to x_{\mathsf{id}}$. Upon input user id, outputs his secret key x_{id}.
- The *Join Oracle* $\mathcal{JO}$: $(\mathsf{id}, \mathsf{ca}) \to cert_{\mathsf{ca}}$. Upon input id of group ca, outputs the $cert_{\mathsf{ca}}$ corresponding to an honest Iss-executing CA.
- The *Issue Oracle* $\mathcal{IO}$: $(\mathsf{id}, \mathsf{ca}) \to cert_{\mathsf{ca}}$. Upon input id of group ca, outputs the $cert_{\mathsf{ca}}$ corresponding to an honest Join-executing user.
- The *Corruption Oracle* $\mathcal{CO}$: $(\mathsf{id}, \mathsf{ca}) \to (x_{\mathsf{id}}, aux_{\mathsf{id}}, cert_{\mathsf{ca}})$. Upon input user id of group ca, outputs the secret keys $(x_{\mathsf{id}}, aux_{\mathsf{id}}, cert_{\mathsf{ca}})$.
- The *Signing Oracle* $\mathcal{SO}$: $(\mathsf{id}, \mathsf{ca}, \mathsf{oa}, M) \to \sigma$. Upon input user id, group ca, oa and a message M, outputs a valid signature.
- The *Open Oracle* $\mathcal{OO}$: $(\mathsf{oa}, \mathsf{ca}, M, \sigma) \to (\mathsf{id}, \omega)$. Upon input a valid signature σ for message M under ca, oa, outputs the signer id and the proof ω.
  Remark: $\mathcal{KEO}_O$ is a stronger oracle than $\mathcal{OO}$ in the sense that $\mathcal{KEO}_O$ directly gives the secret key for OA, while $\mathcal{OO}$ only opens a particular signature.

Anonymity: We have the following **Experiment Anon** for anonymity:

1. Simulator $\mathcal{S}$ invokes Init. $\mathcal{S}$ invokes UKg, Join, Iss together q_u times to generate a set of honest users, denoted HU, with secret keys and certificates.
2. $\mathcal{A}$ queries $\mathcal{RO}, \mathcal{CO}, \mathcal{OO}, \mathcal{IO}, \mathcal{KEO}_c, \mathcal{KEO}_u, \mathcal{KEO}_o$ in arbitrary interleaf.
3. $\mathcal{A}$ selects two users $\mathsf{id}_0, \mathsf{id}_1 \in$ HU, $\mathsf{ca}_g, \mathsf{oa}_g$ a message M and gives them to $\mathcal{S}$. Then $\mathcal{S}$ randomly chooses $b \in \{0, 1\}$ and returns the gauntlet ciphertext $\sigma \leftarrow \mathcal{SO}(\mathsf{id}_b, \mathsf{ca}_g, \mathsf{oa}_g, M)$. oa_g should not be input to $\mathcal{OO}, \mathcal{KEO}_o$ before.
4. $\mathcal{A}$ queries $\mathcal{RO}, \mathcal{CO}, \mathcal{OO}, \mathcal{IO}, \mathcal{KEO}_c, \mathcal{KEO}_u, \mathcal{KEO}_o$ in arbitrary interleaf. oa_g should not be input to $\mathcal{OO}, \mathcal{KEO}_o$.
5. $\mathcal{A}$ delivers an estimate $\hat{b} \in \{0, 1\}$ of b.

$\mathcal{A}$ also has write access to registration table reg in the experiment. $\mathcal{A}$ wins the Experiment Anon if $\hat{b} = b$, and oa_g has never been queried to $\mathcal{KEO}_o$. $\mathcal{A}$'s advantage is its probability of winning Experiment Anon minus half.
Remark: By not allowing to query the gauntlet oa_g, our model is closer to that of [6] which does not support any $\mathcal{OO}$, than to that of [3] which supports $\mathcal{OO}$.

Definition 8. *The IBGS is anonymous if no PPT adversary has a non-negligible advantage in Experiment Anon.*

Traceability: We have the following **Experiment Trace** for traceability:

1. $\mathcal{S}$ invokes Init. $\mathcal{S}$ invokes UKg, Join, Iss together q_u times to generate a set of honest users, denoted HU, with secret keys and certificates.
2. $\mathcal{A}$ queries $\mathcal{RO}, \mathcal{CO}, \mathcal{JO}, \mathcal{KEO}_c, \mathcal{KEO}_u, \mathcal{KEO}_o$ in arbitrary interleaf.
3. $\mathcal{A}$ delivers signature σ for messages M for group ca and open authority oa. ca should not be input to $\mathcal{KEO}_c$.

$\mathcal{A}$ also has read access to reg. $\mathcal{A}$ wins the Experiment Trace if $\mathsf{GVf}(\mathsf{ca}, \mathsf{oa}, M, \sigma) = 1$, either $i = \perp$ or $\mathsf{Judge}(\mathsf{ca}, i, \mathsf{oa}, m, \sigma, \omega) = 0$, where $(i, \omega) \leftarrow \mathsf{Open}(\mathsf{ca}, x_{\mathsf{oa}}, \mathsf{reg}, M, \sigma)$, ca has never been queried to $\mathcal{KEO}_c$, and (i, ca) has never been queried to $\mathcal{CO}$, $\mathcal{A}$'s advantage is its probability of winning.

Definition 9. *The IBGS is traceable if no PPT adversary has a non-negligible advantage in Experiment Trace.*

Non-Frameability: We have the following **Experiment NF** for non-frameability:

1. $\mathcal{S}$ invokes Init. $\mathcal{S}$ invokes UKg, Join, Iss together q_u times to generate a set of honest users, denoted HU, with secret keys and certificates.
2. $\mathcal{A}$ queries $\mathcal{RO}, \mathcal{CO}, \mathcal{SO}, \mathcal{IO}, \mathcal{KEO}_c, \mathcal{KEO}_u, \mathcal{KEO}_o$ in arbitrary interleaf.
3. $\mathcal{A}$ delivers (σ, M, i, ω), where ω is the proof of user i signed the signature σ for messages M with group ca and open authority oa.

$\mathcal{A}$ also has write access to reg. $\mathcal{A}$ wins the Experiment NF if $\mathsf{GVf}(\mathsf{ca}, \mathsf{oa}, M, \sigma) = 1$, $\mathsf{Judge}(\mathsf{ca}, i, \mathsf{oa}, M, \sigma, \omega) = 1$, i has never been queried to $\mathcal{CO}$ and σ is not the output from $\mathcal{SO}$ for $M, i, \mathsf{ca}, \mathsf{oa}$. $\mathcal{A}$'s advantage is its probability of winning.

Definition 10. *The IBGS is non-frameable if no PPT adversary has a non-negligible advantage in Experiment NF.*

Definition 11. *An IBGS scheme is secure if it is correct, anonymous, traceable and non-frameable.*

4 Constructions

In this paper, we present a generic construction for *identity-based group signature (IBGS)* which is applicable to different kinds of relations between the identity based CA, users and open authority. After the generic construction, we give an efficient implementation which is provably secure in the random oracle model.

4.1 Generic construction

A generic *IBGS* is a tuple (Init, OKg, GKg, UKg, Join, Iss, GSig, GVf, Open, Judge) :

– Init, GKg, OKg, UKg, Open, Judge follows the syntax.

- Join,Iss is a pair of interactive protocols with common inputs ca and id. Iss's additional inputs are x_{ca} and aux_{ca}. Join's additional inputs are x_{id} and aux_{id}. Join runs a proof of knowledge protocol to proof that he knows x_{id} and aux_{id} to Iss. At the conclusion, Join obtains $cert_{\mathsf{id}}$ satisfying $((x_{\mathsf{id}}, aux_{\mathsf{id}}, cert_{\mathsf{id,ca}}), \mathsf{id}) \in \mathcal{R}_{C,\mathsf{ca}}$, and Iss stores $(\mathsf{id}, cert_{\mathsf{id,ca}})$ in a registration table reg. Join may also obtain aux_{ca} as part of $cert_{\mathsf{id,ca}}$.
- GSig: (id, ca, oa, x_{id}, aux_{id}, $cert_{\mathsf{id}}$, M) $\mapsto \sigma$. A user id who has $cert_{\mathsf{id}}$ runs:

$$SPK\{(\mathsf{id}, x_{\mathsf{id}}, aux_{\mathsf{id}}, cert_{\mathsf{id,ca}}, r) : (x_{\mathsf{id}}, aux_{\mathsf{id}}, \mathsf{id}) \in \mathcal{R}_U$$
$$\wedge \ (\mathsf{id}, aux_{\mathsf{id}}, cert_{\mathsf{id,ca}}) \in \mathcal{R}_{C,\mathsf{ca}} \ \wedge \ \mathsf{ctxt} = Enc(\mathsf{id}, \mathsf{oa}, r)\}(M)$$

The signature σ is obtained from the above SPK, following [10]'s notion.
- GVf: $(\sigma, M) \mapsto 0$ or 1. On input the signature σ, a verifier verifies σ according to the above SPK. The verifier outputs 1 for valid signature and 0 otherwise.

4.2 An instantiation: IBGS-SDH

We instantiate the generic construction above in the SDH group.
Init: On input the security parameter 1^λ, generates a pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ where the above three (mutiplicative) cyclic groups are of order p. The IM_A (resp. IM_O, IM_U) secret key is $x_A \in \mathbb{Z}_p^*$ (resp. x_O, x_U) and public keys are $g_A, y_A = g_A^{x_A} \in \mathbb{G}_2$ (resp. $g_O, y_O = g_O^{x_O}$, and $g_U, y_U = g_U^{x_U}$). Let u be a generator in $\mathbb{G}_1$. Define cryptographic hash functions $\mathcal{H}_A : \{0,1\}^* \to \mathbb{Z}_p^*$, $\mathcal{H}_U : \{0,1\}^* \to \mathbb{G}_1$, $\mathcal{H}_O : \{0,1\}^* \to \mathbb{G}_1$, $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p^*$.

For CA, define $\mathcal{R}_A = \{((x_{\mathsf{ca}}, R), \mathsf{ca}) : g_A^{x_{\mathsf{ca}}} = R y_A^{\mathcal{H}_A(R||\mathsf{ca})}\}$. For OA, define $\mathcal{R}_O = \{(x_{\mathsf{oa}}, \mathsf{oa}) : x_{\mathsf{oa}} = \mathcal{H}_O(\mathsf{oa})^{x_O}\}$. For user, define $\mathcal{R}_U = \{(x, i) : x = \mathcal{H}_U(i)^{x_U}\}$. For certificate, define $\mathcal{F} = \{\langle \mathcal{R}_{C,i} \rangle : i\}$ with trapdoor x_i. $\mathcal{R}_{C,\mathsf{ca}} = \{(\mathsf{id}, (A, e)) : A^{e + x_{\mathsf{ca}}} \mathcal{H}_U(\mathsf{id}) = u\}$.

Let $g_0, g_1, g_2, g_3, g_4, u$ are generators in $\mathbb{G}_1$. Then:
$\mathsf{param} = (\hat{e}, g_A, y_A, g_O, y_O, g_U, y_U, g_0, ..., g_4, u, \mathcal{H}_A, \mathcal{H}_U, \mathcal{H}_O, \mathcal{H}, \mathcal{R}_A, \mathcal{R}_U, \mathcal{R}_O, \mathcal{F})$.

OKg: On input OA identity oa, the identity manager IM_O uses x_O to compute OA secret key $x_{\mathsf{oa}} = \mathcal{H}_O(\mathsf{oa})^{x_O}$.

GKg: On input CA identity ca, the identity manager IM_A defines $\mathcal{R}_{C,\mathsf{ca}} = \{(\mathsf{id}, (A, e)) : A^{e + x_{\mathsf{ca}}} \mathcal{H}_U(\mathsf{id}) = u\}$ and computes as follows:

1. Randomly generate $r \in \mathbb{Z}_p^*$.
2. Compute $aux_{\mathsf{ca}} = g_A^r$, $x_{\mathsf{ca}} = r + \mathcal{H}_A(aux_{\mathsf{ca}}||\mathsf{ca}) x_A \bmod p$.

This is taken from BNN-IBI [2]. Finally CA gets $(x_{\mathsf{ca}}, aux_{\mathsf{ca}})$.

UKg: On input user identity id, the identity manager IM_U uses x_U to compute user secret key $x_{\mathsf{id}} = \mathcal{H}_U(\mathsf{id})^{x_U}$.

Join,Iss: Common inputs are id, ca. Join's additional input is x_{id} and Iss's additional inputs are $x_{\mathsf{ca}}, aux_{\mathsf{ca}}$. Join firstly runs a proof of knowledge of x_{id} for id.

Then Iss uses x_{ca}, aux_{ca} to computes $cert_{\mathsf{id,ca}} = (A, e)$ satisfying $(\mathsf{id}, cert_{\mathsf{id,ca}}) \in \mathcal{R}_{C,\mathsf{ca}}$. Iss randomly selects $e \in \mathbb{Z}_p^*$, and computes $A = (u/\mathcal{H}_U(\mathsf{id}))^{1/(e+x_{\mathsf{ca}})}$. Iss sends $(A, e, aux_{\mathsf{ca}})$ to Join. The validity of aux_{ca} can be checked by BNN's IBI [2]. Note u is a fairly generated public parameter, Join accepts the certificate if and only if $\hat{e}(u, g_A) = \hat{e}(A, g_A)^e \hat{e}(A, S) \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A)$, where $S = g_A^{x_{\mathsf{ca}}} = aux_{\mathsf{ca}} y_A^{\mathcal{H}_A(aux_{\mathsf{ca}}||\mathsf{ca})}$. Finally Join obtains $cert_{\mathsf{id,ca}}, aux_{\mathsf{ca}}$. Iss computes $W = \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A)$, and puts (id, A, e, W) in reg.

GSig: A member id from group ca with secret key x and certificate (A, e) computes a signature σ for message M and oa by

$$\begin{aligned} SPK\{(\mathsf{id}, x, (A, e), d) : x = \mathcal{H}_U(\mathsf{id})^{x_U} \ \wedge \ A^{e+x_{\mathsf{ca}}} \mathcal{H}_U(\mathsf{id}) = u \\ \wedge \ \mathsf{ctxt} = \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A) \hat{e}(\mathcal{H}_O(\mathsf{oa}), y_O)^d \ \wedge \ U = g_O^d\}(M) \end{aligned} \quad (1)$$

which is equivalent to

$$\begin{aligned} SPK\{(\mathsf{id}, x, (A, e), d) : \hat{e}(x, g_U) = \hat{e}(\mathcal{H}_U(\mathsf{id}), y_U) \\ \wedge \ \hat{e}(u, g_A) = \hat{e}(A, g_A)^e \hat{e}(A, S) \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A) \quad (2) \\ \wedge \ \mathsf{ctxt} = \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A) \hat{e}(\mathcal{H}_O(\mathsf{oa}), y_O)^d \ \wedge \ U = g_O^d \\ \wedge \ S = aux_{\mathsf{ca}} y_A^{\mathcal{H}_A(aux_{\mathsf{ca}}||\mathsf{ca})}\}(M) \quad (3) \end{aligned}$$

The further instantiation is as follows. Randomly selects $s_1, d \in \mathbb{Z}_p^*$. Computes $s_2 = es_1$. The masked images are:

$$t_0 = g_0^{s_1} \ \wedge \ t_1 = x g_1^{s_1} \ \wedge \ t_2 = \mathcal{H}_U(\mathsf{id}) g_2^{s_1} \ \wedge \ t_3 = A g_3^{s_1} \ \wedge \ t_5 = t_3^e g_4^{s_1} \quad (4)$$

And we have: $\mathsf{ctxt} = \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A) \hat{e}(\mathcal{H}_O(\mathsf{oa}), y_O)^d \ \wedge \ U = g_O^d$.

Randomly selects $r_1, r_2, r_3, r_4 \in \mathbb{Z}_p$, $R_1, R_2, R_3 \in \mathbb{G}_1$. Computes:

$$\begin{aligned} \tau_0 = g_0^{r_1} \ \wedge \ \tau_1 = R_1 g_1^{r_1} \ \wedge \ \tau_2 = R_2 g_2^{r_1} \ \wedge \ \tau_3 = R_3 g_3^{r_1} \\ \wedge \ \tau_4 = [\hat{e}(g_1, g_U)^{-1} \hat{e}(g_2, y_U)]^{r_1} \ \wedge \ \tau_5 = t_3^{r_3} g_4^{r_1} \\ \wedge \ \tau_6 = \hat{e}(g_3, g_A)^{r_2} [\hat{e}(g_3, S) \hat{e}(g_2 g_4, g_A)]^{r_1} \ \wedge \ \tau_7 = g_A^{r_4} \\ \wedge \ \tau_8 = \hat{e}(\mathcal{H}_O(\mathsf{oa}), y_O)^{r_4} \hat{e}(g_2, g_A)^{-r_1} \end{aligned}$$

The challenge is:

$$c = \mathcal{H}((t_0, \cdots, t_3, t_5)||(\tau_0, \cdots, \tau_8)||aux_{\mathsf{ca}}||\mathsf{ctxt}||U||M) \quad (5)$$

The responses are:

$$\begin{aligned} z_0 = r_1 - cs_1 \ \wedge \ Z_1 = R_1 x^{-c} \ \wedge \ Z_2 = R_2 \mathcal{H}_U(i)^{-c} \\ \wedge \ Z_3 = R_3 A^{-c} \ \wedge \ z_4 = r_3 - ce \ \wedge \ z_5 = r_2 - cs_2 \ \wedge \ z_6 = r_4 - cd \end{aligned}$$

The signature σ is: $(t_0, \cdots, t_3, t_5)||c||(z_0, \cdots, z_6)||aux_{\mathsf{ca}}||\mathsf{ctxt}||U||M$.

GVf: Given a signature σ, it computes:

$$\begin{aligned} t_4 = \hat{e}(t_1, g_U)^{-1}\hat{e}(t_2, y_U) \;\wedge\; t_6 = \hat{e}(u, g_A)^{-1}\hat{e}(t_2 t_5, g_A)\hat{e}(t_3, S) \\ \wedge\; t_8 = \mathsf{ctxt} \cdot \hat{e}(t_2, g_A)^{-1} \;\wedge\; \tau_0 = g_0^{z_0} t_0^c \;\wedge\; \tau_1 = Z_1 g_1^{z_0} t_1^c \\ \wedge\; \tau_2 = Z_2 g_2^{z_0} t_2^c \;\wedge\; \tau_3 = Z_3 g_3^{z_0} t_3^c \;\wedge\; \tau_4 = [\hat{e}(g_1, g_U)^{-1}\hat{e}(g_2, y_U)]^{z_0} t_4^c \\ \wedge\; \tau_5 = t_3^{z_4} g_4^{z_0} t_5^c \;\wedge\; \tau_6 = \hat{e}(g_3, g_A)^{z_5}[\hat{e}(g_3, S)\hat{e}(g_2 g_4, g_A)]^{z_0} t_6^c \;\wedge\; \tau_7 = g_A^{z_6} U^c \\ \wedge\; \tau_8 = \hat{e}(\mathcal{H}_O(\mathsf{oa}), y_O)^{z_6}\hat{e}(g_2, g_A)^{-z_0} t_8^c \;\wedge\; S = aux_{\mathsf{ca}} y_A^{\mathcal{H}_A(aux_{\mathsf{ca}}||\mathsf{ca})} \end{aligned} \tag{6}$$

Then it computes challenge $\hat{c}$ according to Eq. (5), and compares it to the challenge c received in the signature. If they are equal, output 1 for *valid signature*. In all other cases, output 0.

Open: The open authority uses his secret key x_{oa} to open the encryption in the signature σ. Denote $Q_{\mathsf{oa}} = \mathcal{H}_O(\mathsf{oa})$. He computes:

$$m = \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A) = \mathsf{ctxt}/\hat{e}(x_{\mathsf{oa}}, U)$$

The open authority compares W with the registration table reg. If no such entry is find, output $\perp$. If it is found to be user id, the open authority computes a proof of knowledge of x_{oa} such that $\hat{e}(x_{\mathsf{oa}}, U) = \mathsf{ctxt}/m$:

1. Randomly picks $s'_0 \in \mathbb{Z}_p$. Computes:
$t'_0 = x_{\mathsf{oa}} h^{s'_0} \;\wedge\; t'_1 = \hat{e}(h, U)^{s'_0} \;\wedge\; t'_2 = \hat{e}(h, g_O)^{s'_0}$.
2. Randomly picks $r'_0, r'_1 \in \mathbb{Z}_p$. Computes:
$\tau'_0 = Q_{\mathsf{oa}}^{r'_1} h^{r'_0} \;\wedge\; \tau'_1 = \hat{e}(h, U)^{r'_0} \;\wedge\; \tau'_2 = \hat{e}(h, g_O)^{r'_0}$.
3. Computes $c' = \mathcal{H}((t'_0, t'_1, t'_2)||(\tau'_0, \tau'_1, \tau'_2)||\mathsf{ctxt}||U||m)$.
4. Computes $z'_0 = r'_0 - c's'_0$, $Z'_1 = Q_{\mathsf{oa}}^{r'_1} x_{\mathsf{oa}}^{c'}$.

Outputs the proof $\omega = (t'_0||c'||(z'_0, Z'_1))$ to judge.

Judge: On input $\mathsf{id}, \mathsf{ca}, \mathsf{oa}$, the signature σ and the proof ω, it computes:

$$m = \hat{e}(\mathcal{H}_U(\mathsf{id}), g_A) \;\wedge\; m' = \mathsf{ctxt}/m \;\wedge$$
$$t'_1 = \hat{e}(t'_0, U)/m' \;\wedge\; t'_2 = \hat{e}(t'_0, g_O)\hat{e}(Q_{\mathsf{oa}}, y_O) \;\wedge \tag{7}$$
$$\tau'_0 = Z'_1 {t'_0}^{c'} h^{z'_0} \;\wedge\; \tau'_1 = \hat{e}(h, U)^{z'_0} {t'_1}^{c'} \;\wedge\; \tau'_2 = \hat{e}(h, g_O)^{z'_0} {t'_2}^{c'} \tag{8}$$

Then compares if $c' = \mathcal{H}((t'_0, t'_1, t'_2)||(\tau'_0, \tau'_1, \tau'_2)||\mathsf{ctxt}||U||m)$. If it is true, output 1. Otherwise, output 0.

5 Security Theorems

We now give the security theorems for the above instantiation. It follows the definition in section 3. The proofs can be found in the full version of this paper.

Theorem 2. *The IBGS-SDH scheme is correct.*

Proof. Obvious.

Theorem 3. *The IBGS-SDH is anonymous in the random oracle model if and only if the DDH Assumption in* $\mathbb{G}_1$ *and the co-DBDH Assumption in* $(\mathbb{G}_2, \mathbb{G}_1)$ *both hold.*

Proof. Suppose $\mathcal{A}$ is a PPT algorithm that breaks the anonymity of the group signature. Then we show how to construct a PPT algorithm $\mathcal{S}$ that solves the Lockstep DDH+coDBDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$, which is equivalent to the co-DBDH problem in $(\mathbb{G}_2, \mathbb{G}_1)$ and the DDH problem in $\mathbb{G}_1$.

$\mathcal{S}$ is given the instance $g'_1, {g'_1}^{\alpha}, {g'_1}^{\beta_i}, {g'_1}^{\gamma_i} \in \mathbb{G}_1$ for $1 \leq i \leq 4$; $g'_2, {g'_2}^{\delta_1}, {g'_2}^{\delta_2} \in \mathbb{G}_2$ and $R \in \mathbb{G}_T$ for unknown $\alpha_i, \beta_i, \delta_1, \delta_2 \in \mathbb{Z}_p$. $\mathcal{S}$ is sets the public parameter $g_O = g'_2, y_O = {g'_2}^{\delta_1}, g_0 = g'_1, g_1 = {g'_1}^{\beta_1}, g_1 = 2 = {g'_1}^{\beta_2}, g_3 = {g'_1}^{\beta_3}, g_4 = {g'_1}^{\beta_4}$. $\mathcal{S}$ generates $g_A, x_A, y_A = g_A^{x_A}$, $g_U, x_U, y_U = g_U^{x_U}$ and $u = g_A$. $\mathcal{S}$ randomly picks $\ell \in \{1, ..., q_H\}$, where q_H is the number of query to $\mathcal{H}_O$. $\mathcal{S}$ provides $\mathcal{A}$ the parameters param.

The oracles are simulated as follows:

- $\mathcal{H}$ is random oracle.
- $\mathcal{H}_A(aux_i||i)$: On input new aux_i, i, randomly pick $\lambda \in \mathbb{Z}_p$ Return λ. Store (aux_i, i, λ) in tape $\mathcal{L}_A$.
- $\mathcal{H}_U(i)$: On input new i, randomly pick $\lambda \in \mathbb{Z}_p$ and return g_U^{λ}. Store (i, λ) in tape $\mathcal{L}_U$.
- $\mathcal{H}_O(i)$: On input new i, randomly pick $\lambda \in \mathbb{Z}_p$ and return g_O^{λ}. Store (i, λ) in tape $\mathcal{L}_O$. For the ℓ-th query, return $Q = g'_1$ and back patch (i, Q) in $\mathcal{L}_O$. Denote this identity as i_g.
- $\mathcal{KEO}_u(i)$: Computes $\mathcal{H}_U(i)$. Then $x_i = y_U^{\lambda}$, where $(i, \lambda) \in \mathcal{L}_U$.
- $\mathcal{KEO}_c(ca)$: On input ca, randomly pick $h, x_{ca} \in \mathbb{Z}_p$ and computes $aux_{ca} = g_A^{x_{ca}} y_A^{-h}$. $\mathcal{S}$ back patches $\mathcal{H}_A(aux_{ca}||ca) = h$. Store (aux_{ca}, ca, h) in tape $\mathcal{L}_A$. Return (x_{ca}, aux_{ca}).
- $\mathcal{KEO}_o(oa)$: Computes $\mathcal{H}_O(oa)$. Then $x_{oa} = y_O^{\lambda}$, where $(oa, \lambda) \in \mathcal{L}_O$. If $oa = i_g$, declare failure and exit.
- $\mathcal{IO}(i, ca)$: It interacts with the honest user i. Computes (x_{ca}, aux_{ca}) as in $\mathcal{KEO}_c(ca)$. Randomly selects $e \in \mathbb{Z}_p$, and computes $A = (u/\mathcal{H}_U(i))^{1/(e+x_{ca})}$, $W = \hat{e}(\mathcal{H}_U(i), g_A)$. Stores (i, A, e, W) in reg. Returns (A, e, aux_{ca}) to honest user i.
- $\mathcal{CO}(i, ca)$: On input the identity, this oracle outputs the user's secret keys. Computes $\mathcal{H}_1(i)$. Computes x_i as in $\mathcal{KEO}_u(i)$. Computes $cert_{i,ca}$ as in $\mathcal{IO}(i, ca)$. Returns $(x_i, cert_{i,ca})$.
- $\mathcal{OO}(\mathsf{oa}, \mathsf{ca}, m, \sigma)$: Computes $\mathcal{H}_1(\mathsf{oa})$. Then $x_{\mathsf{oa}} = y_A^{\lambda}$, where $(\mathsf{oa}, \lambda) \in \mathcal{L}_1$. Return $(i, \omega) \leftarrow \mathsf{Open}(\mathsf{ca}, x_{\mathsf{oa}}, \mathsf{reg}, m, \sigma)$. If $oa = i_g$, declare failure and exit.

At any time, $\mathcal{A}$ can query the oracles above. At some point, $\mathcal{A}$ sends the gauntlet identity i_0, i_1, group ca, open authority oa and message M to $\mathcal{S}$. $\mathcal{S}$ flips a coin $b \in \{0, 1\}$ and computes $(x_b, A_b, e_b) \leftarrow \mathcal{CO}(i_b, ca)$. $\mathcal{S}$ sets $t_0 = {g'_1}^{\alpha}, t_1 = x_b {g'_1}^{\gamma_1}, t_2 = \mathcal{H}_U(i_b){g'_1}^{\gamma_2}, t_3 = A_b {g'_1}^{\gamma_3}, t_5 = t_3^{e_b} {g'_1}^{\gamma_4}$. $\mathcal{S}$ randomly chooses a challenge $c \in \mathbb{Z}_p$ and response $z_0, ..., z_6$ from suitable domains. It computes $\tau_0, ..., \tau_8$ as in GVf. $\mathcal{S}$ sets $U = {g'_2}^{\delta_2}$ and computes $\mathsf{ctxt} = \hat{e}(\mathcal{H}_U(i_b), g_A)R$. Then back patch c to $\mathcal{H}$ as Eq. 5. $\mathcal{S}$ returns the signature σ_g as the gauntlet to $\mathcal{A}$.

$\mathcal{A}$ finally outputs a bit $\hat{b}$. If $\hat{b} = b$, $\mathcal{S}$ returns "yes" for the Lockstep DDH+coDBDH problem. Otherwise, $\mathcal{S}$ returns "no". By the back patch above, if $\mathcal{A}$ has a non-negligible advantage ε in winning the game, $\mathcal{S}$ has advantage ε/q_H in solving the Lockstep DDH+coDBDH problem, and hence can either solve the DDH problem in $\mathbb{G}_1$ or the co-DBDH problem in $(\mathbb{G}_2, \mathbb{G}_1)$.

Bow we derive the opposite reduction in the Theorem statement: Give the Adversary a Lockstep DDH+coDBDH Oracle which can solve the Lockstep DDH+coDBDH Problem, and then show it can crack Anonymity. If the adversary is given a signature $\sigma = (t_0, \cdots, t_3, t_5)||c||(z_0, \cdots, z_6)||aux_{\mathsf{ca}}||\mathsf{ctxt}||U||M$ which can pass GVf. $\mathcal{A}$ is also given (x_b, A_b, e_b) for users id_b where $b \in \{0,1\}$. Then $\mathcal{A}$ randomly flips a coin $b = 0/1$ and inputs to the Oracle: $g_1' = g_0$, ${g_1'}^{\alpha} = t_0$, ${g_1'}^{\beta_1} = g_1$, ${g_1'}^{\gamma_1} = t_1/x_b$, ${g_1'}^{\beta_2} = g_2$, ${g_1'}^{\gamma_2} = t_2/\mathcal{H}_U(id_b)$, ${g_1'}^{\beta_3} = g_3$, ${g_1'}^{\gamma_3} = t_3/A_b$, ${g_1'}^{\beta_4} = g_4$, ${g_1'}^{\gamma_4} = t_5/t_3^{e_b}$, $g_2' = g_O$, ${g_2'}^{\delta_1} = y_O$, ${g_2'}^{\delta_2} = U$, $R = \mathsf{ctxt}/\hat{e}(\mathcal{H}_U(id_b), g_A)$. If the Oracle outputs 1, then $\mathcal{A}$ outputs id_b as the signer. Otherwise, $\mathcal{A}$ outputs id_{1-b} as the signer. □

Theorem 4. *The IBGS-SDH is traceable in the random oracle model if and only if the k-CAA2 assumption holds.*

Proof. Let $\mathcal{A}$ be a PPT adversary attacking the traceability. We show that given a colluding group of k signers, with the knowledge of the opening key and access to some oracles, we can use $\mathcal{A}$ to solve the k-CAA2 problem.

$\mathcal{S}$ is given the tuple $u, v \in \mathbb{G}_1$, $g_2, {g_2}^{\gamma} \in \mathbb{G}_2$ and pairs (A_i, e_i, λ_i) with distinct and nonzero e_i's satisfying $A_i^{\gamma+e_i} v^{\lambda_i} = u$ for $1 \leq i \leq k$ as input. The value $s = log_u(v)$ is also given to $\mathcal{S}$.

$\mathcal{S}$ sets $g_A = g_2, g_U = v$. $\mathcal{S}$ randomly selects $x_A, y_A = g_A^{x_A}$, $x_U, y_U = g_U^{x_U}$ and $g_O, x_O, y_O = g_O^{x_O}$. $\mathcal{S}$ randomly selects μ and sets $g_3 = v^{\mu}$. $\mathcal{S}$ setups the rest of param and provides to $\mathcal{A}$. $\mathcal{S}$ randomly picks $\ell \in \{1, ..., q_c\}$, where q_c is number of query to $\mathcal{CO}$.

The oracles are simulated as follows:

- $\mathcal{H}_U(i)$: On input new i, randomly pick λ_j from the given k-CAA2 tuple and return v^{λ_j}. Store (i, λ_j) in tape $\mathcal{L}_U$.
- $\mathcal{JO}(i, ca)$: It interacts with honest issuer ca. Computes x_i as in $\mathcal{KEO}_u(i)$. Then interacts with ca with x_i. Finally ca returns $cert_{i,ca}$.
- $\mathcal{CO}(i, ca)$: On input the identity, this oracle outputs the user's secret keys. Computes x_i as in $\mathcal{KEO}_u(i)$. Computes (x_{ca}, aux_{ca}) as in $\mathcal{KEO}_c(ca)$. Randomly selects $e \in \mathbb{Z}_p$, and computes $A = (u/\mathcal{H}_U(i))^{1/(e+x_{ca})}$, $W = \hat{e}(\mathcal{H}_U(i), g_A)$. Stores (i, A, e, W) in reg. Returns (x_i, A, e, aux_{ca}).
  For the ℓ-th query, randomly selects $h \in \mathbb{Z}_p$ and computes $aux_{ca} = g_2^{\gamma} y_A^{-h}$. $\mathcal{S}$ back patch $\mathcal{H}_A(aux_{ca}||ca) = h$. Picks a pair of (A_i, e_i, λ_i) from the k-CAA2 tuple. Back patches (i, λ_i) to $\mathcal{L}_U$. Then we have $x_i = y_U^{\lambda_i}$. Returns $(x_i, A_i, e_i, aux_{ca})$. Computes $W = \hat{e}(\mathcal{H}_U(i), g_A)$. Stores (i, A_i, e_i, W) in reg. Denote this identity as ca_g. If $ca = \mathsf{ca}_g$ in future queries, also runs the above steps.

Other oracles are similar to the proof of theorem 3. ca_g should not be input to the $\mathcal{KEO}_c$. Suppose $\mathcal{A}$ can output a valid signature σ such that the OA cannot trace the identity of the signer, or the OA can find the identity of the signer but cannot prove that to Judge.

Below we proof the soundness of the proof system between Open and Judge. Rewind the simulation to obtain:

$$1 = \Delta Z_1' h^{\Delta z_0'} {t_0'}^{\Delta c'} \ \wedge \ 1 = \hat{e}(h,U)^{\Delta z_0'} {t_1'}^{\Delta c'} \ \wedge \ 1 = \hat{e}(h,g_O)^{\Delta z_0'} {t_2'}^{\Delta c'}$$
$$t_0' = {\Delta Z_1'}^{1/\Delta c'} h^{\Delta z_0'/\Delta c'} \ \wedge \ t_1' = \hat{e}(h,U)^{\Delta z_0'/\Delta c'} \ \wedge \ t_2' = \hat{e}(h,g_O)^{\Delta z_0'/\Delta c'}$$

And notice that we have:

$$t_1' = \hat{e}(h,U)^{s_0'} = \hat{e}(t_0',U){m'}^{-1}$$
$$t_2' = \hat{e}(h,g_O)^{s_0'} = \hat{e}(t_0' {x_{\mathsf{oa}}}^{-1}, g_O)$$

Let $\tilde{s_0'} = -\Delta z_0'/\Delta c'$. Hence $m' = \hat{e}(t_0',U){t_1'}^{-1} = \hat{e}(h^{\tilde{s_0'}} t_0', U)$. Since we have $t_0' {x_{\mathsf{oa}}}^{-1} = h^{-\tilde{s_0'}}$, then $m' = \hat{e}(h^{\tilde{s_0'}} t_0', U) = \hat{e}(x_{\mathsf{oa}}, U)$. Therefore we extract the witness $x_{\mathsf{oa}} = t_0' h^{\tilde{s_0'}}$. Hence for an OA with secret key x_{oa}, he can always output a valid proof to the Judge if he knows the identity of the signer.

If finally $\mathcal{A}$ returns a signature with group $\mathsf{ca} = \mathsf{ca}_g$, then we rewind the simulation to the point where c is computed.

After rewind, we get: $g_0^{\Delta z_0} t_0^{\Delta c} = 1$, $\Delta Z_1 g_1^{\Delta z_0} t_1^{\Delta c} = 1$, $\Delta Z_2 g_2^{\Delta z_0} t_2^{\Delta c} = 1$, $\Delta Z_3 g_3^{\Delta z_0} t_3^{\Delta c} = 1$, $t_3^{\Delta z_4} t_5^{\Delta c} = 1$, $g_A^{\Delta z_6} U^{\Delta c} = 1$.

Let $\tilde{s}_1 = -\Delta z_0/\Delta c$, $\tilde{x} = \Delta Z_1^{-1/\Delta c}$, $\tilde{\mathcal{H}} = \Delta Z_2^{-1/\Delta c} = \mathcal{H}_1(i)$, $\tilde{A} = \Delta Z_3^{-1/\Delta c}$, $\tilde{e} = -\Delta z_4/\Delta c$, $\tilde{s}_2 = -\Delta z_5/\Delta c$, $\tilde{d} = -\Delta z_6/\Delta c$. We have:

$$\begin{aligned} \hat{e}(g_3,g_A)^{\Delta z_5}[\hat{e}(g_3,S)\hat{e}(g_2,g_A)]^{\Delta z_1} t_6^{\Delta c} &= 1 \\ \hat{e}(g_3,g_A)^{\tilde{s}_2}[\hat{e}(g_3,S)\hat{e}(g_2,g_A)]^{\tilde{s}_1} &= t_6 \\ &= \hat{e}(u,g_A)^{-1}\hat{e}(t_2 t_5, g_A)\hat{e}(t_3,S). \end{aligned}$$

After rearranging, we have:

$$\hat{e}(u,g_A) = \hat{e}(\tilde{A},g_A)^{\tilde{e}} \hat{e}(\tilde{A},S)\hat{e}(\tilde{\mathcal{H}},g_A) e(g_3,g_A)^{\tilde{e}\tilde{s}_1 - \tilde{s}_2}$$

If $\tilde{e}\tilde{s}_1 = \tilde{s}_2$, then we get a pair of $(\tilde{A},\tilde{e},\tilde{\mathcal{H}})$ which satisfy $\tilde{A}^{\tilde{e}+\gamma}\tilde{\mathcal{H}} = u$. Then we have $(\tilde{A},\tilde{e},\lambda)$, where $(i,\lambda) \in \mathcal{L}_1$, that solves the k-CAA2 problem.

If $\tilde{e}\tilde{s}_1 \neq \tilde{s}_2$, then we have $\tilde{A}^{\tilde{e}+\gamma}\tilde{\mathcal{H}} g_3^{(\tilde{e}\tilde{s}_1 - \tilde{s}_2)} = u$. Then we have $\lambda^* = \lambda + \mu(\tilde{e}\tilde{s}_1 - \tilde{s}_2)$, where $(i,\lambda) \in \mathcal{L}_1$, such that $(\tilde{A},\tilde{e},\lambda^*)$ solves the k-CAA2 problem.

Hence if $\mathcal{A}$ has a non-negligible advantage ε in winning the game, $\mathcal{S}$ has advantage ε/q_c in solving the k-CAA2 problem.

Now we derive the opposite reduction in the Theorem statement: Give the Adversary a k-CAA2 oracle, and then use it to compute/forge an additional signature, which is not traceable, after k queries to the $\mathcal{CO}$ Oracle. Then $\mathcal{A}$ gets k sets of (A_i, e_i, x_i) for id_i where $1 \leq i \leq k$. $\mathcal{A}$ inputs (A_i, e_i, λ_i), where $(id_i, \lambda_i) \in \mathcal{L}_1$, to the k-CAA2 oracle. The oracle returns a new pair (A_*, e_*, λ_*). $\mathcal{A}$ backpatches (id_*, λ_*) to $\mathcal{L}_1$. $\mathcal{A}$ uses the $\mathcal{KEO}_U$ to find x_* for id_*. Then $\mathcal{A}$ uses (A_*, e_*, x_*) to compute a signature for message m. Then an honest OA will find that the signature is valid and opens to a value $\hat{e}(\mathcal{H}_U(id_*), g_A)$. As it is not in reg, the OA will outputs $\perp$. Hence $\mathcal{A}$ can forge a signature. □

Theorem 5. *The IBGS-SDH is non-frameable in the random oracle model if and only if the co-CDH assumption holds.*

Proof. Assume $\mathcal{A}$ can win Experiment NY with advantage ϵ, and it delivers signature σ, message M and a proof ω to signer i_g. It remains to prove that (1) the VE (Verifiable Encryption) part, ω, validly opens to $\hat{e}(\mathcal{H}_U(i_g), g_A)$; and (2) the signature part is sound.

(1) This means if $\mathsf{Judge}(ca, i_g, oa, M, \sigma, \omega) = 1$, then $\mathsf{ctxt} = \hat{e}(\mathcal{H}_U(i_g), g_A)\hat{e}(Q_B^{x_O}, U)$, where $Q_B = \mathcal{H}_O(oa)$. We prove by forking simulation. Some may find this proof approach not rigorous enough. But this is the state-of-the-art proof technique for the correctness of decryption in many results on VE. Besides, it is possible to modify the security model somewhat slightly to make this kind of proof rigorous. We omit details of the modification for the simplicity of presentation.

Suppose $\mathcal{S}$ is given (P, P^α, Q). $\mathcal{S}$ sets $g_U = P, y_U = P^\alpha$ for the identity manager of members.

$\mathcal{S}$ sets $g_A = g_O = P$. $\mathcal{S}$ randomly selects x_A, x_O and computes $y_A = g_A^{x_A}, y_O = g_O^{x_O}$. $\mathcal{S}$ setups the rest of param and provides to $\mathcal{A}$. $\mathcal{S}$ randomly picks ℓ in $\{1, ..., q_H\}$, where q_H is the number of query to $\mathcal{H}_U$.

The oracles are simulated as follows:

- $\mathcal{H}_U(i)$: On input new i, randomly pick $\lambda \in \mathbb{Z}_p$ and return g_U^λ. Store (i, λ) in tape $\mathcal{L}_U$. For the ℓ-th query, return Q and back patch (i, Q) in $\mathcal{L}_U$. Denote this identity as i_g.
- $\mathcal{KEO}_u(i)$: Computes $\mathcal{H}_U(i)$. Then $x_i = y_U^\lambda$, where $(i, \lambda) \in \mathcal{L}_U$. If $i = i_g$, declares failure and exits.
- $\mathcal{CO}(i, ca)$: On input the identity, this oracle outputs the user's secret keys. Computes $\mathcal{H}_1(i)$. Computes $x_i = y_U^\lambda$, where $(i, \lambda) \in \mathcal{L}_U$. Computes $cert_{i,ca}$ as in $\mathcal{IO}(i, ca)$. Returns $(x_i, cert_{i,ca})$. If $i = i_g$, declares failure and exits.
- $\mathcal{SO}(i, ca, M)$: If $i \neq i_g$, computes $x_i, cert_{i,ca}$ as in $\mathcal{CO}$. Then uses $x_i, (A_i, e_i)$ to sign the message M. Return the signature σ.
  If $\mathcal{SO}(i_g, ca, M)$ is called, randomly selects $t_0, ..., t_3, t_5 \in \mathbb{G}_1$, chooses a challenge $c \in \mathbb{Z}_p$ and response $z_0, ..., z_6$ from suitable domains. It computes $\tau_0, ..., \tau_8$ as in GVf. Then back patch c to $\mathcal{H}$ as Eq. 5. Obviously this signature will pass GVf.

Finally if $\mathcal{A}$ can frame a member i^* of signing a message m, it has probability $1/q_H$ of framing user i_g. $\mathcal{A}$ should not query $\mathcal{CO}(i^*)$ or $\mathcal{SO}(i^*, m)$. If $i^* = i_g$, $\mathcal{S}$ opens the signature and extracts x_{i_g} as the solution to the co-CDH problem.

(2) This means the soundness of the proof system in Equation 3 when ctxt and U are discarded. This is proved in theorem 4.

Hence if $\mathcal{A}$ has a non-negligible advantage ε in winning the game, $\mathcal{S}$ has advantage ε/q_H in solving the co-CDH problem.

Now we derive the opposite reduction in the Theorem statement: Give the Adversary a co-CDH oracle, and then show it can frame. Suppose $\mathcal{A}$ wants to frame user id_*. $\mathcal{A}$ then inputs to the co-CDH oracle: $P = g_U$, $P^\alpha = y_U$, $Q = \mathcal{H}_U(id_*)$. Denote the oracle output Q^α be x_*. Then $\mathcal{A}$ uses x_* to act as an honest user to interact with the Issue Oracle. The oracle outputs a valid certificate (A_*, e_*) for user id_*. Then $\mathcal{A}$ uses (A_*, e_*, x_*) to output a signature σ for message m. After that $\mathcal{A}$ extract the secret key of OA by $\mathcal{KEO}_O$. $\mathcal{A}$ uses it to compute a proof ω that shows id_* signs $sigma$. Hence ω must pass Judge. Then $\mathcal{A}$ outputs $(\sigma, m, id_*, \omega)$ to frame user id_*. □

Summarizing, we have:

Theorem 6. *Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be a pairing. The IBGS-SDH is secure if and only if the DDH Assumption in $\mathbb{G}_1$, the co-DBDH Assumption in $(\mathbb{G}_2, \mathbb{G}_1)$, the k-SDH' Assumption, and the co-CDH Assumption all hold in the random oracle model.*

6 Discussions

6.1 Other instantiation

For the above generic construction, we use a discrete logarithm type of identity based key pairs for CA and pairing type of identity based key pairs for OA and group members to give an instantiation. From [2], we have three identity based identification for discrete logarithm type: Beth [4], Okamoto [21], BNN [2]. They are suitable for constructing the key pairs for both CA and group members. We have different identity based identification for pairing type ([24], [16], [12]). They are suitable for constructing the key pairs for group members. For OA, the key pairs can be obtained from secure identity based encryption which allows efficient verification. Therefore we can form different identity based group signature using different combination of the above key pairs.

For other kinds of certificates in group signature schemes, CA in Ateniese et al [1] has private key (p', q') from the strong RSA assumption. However no existing identity based identification has this form of user key pairs. For Dodis et al [15], there is no CA and the group public key is some accumulated value. Both are not suitable for having identity based group manager.

If one wants the encryption scheme for the open authority to be CCA-2, then we can modify our scheme as follows. We perform the SPK without encryption, and then perform a verifiable encryption scheme from Camenisch and Damgård [9] with Fiat-Shamir heuristic. The encryption scheme used is FullIdent from Boneh and Franklin [7], which is CCA-2. However, the signature size of this scheme will depend on the group size.

6.2 Short Ring signature

We can formulate our group signature scheme without open authority. We refer this kind of signature scheme as ring signature, as the anonymity of the signature scheme is non-revokeable. It extends the idea of ring signature in [23].

Without the open authority, our signature scheme has signature size independent of the group size. To turn the identity based group signature to a short identity based ring signature scheme, we only have to remove the encryption from GSig. The OA, Open, Judge are also removed. Then short identity based ring signature is constructed.

7 Conclusion

In this paper, we present a fully identity based group signature scheme, with identity based group manager, identity based group members and identity based open authority. We give a generic construction and also an instantiation, which the signature size is independent of the group size. We prove the security of the instantiation in the random oracle model. We also showed that a short identity based ring signature can be formed similarly.

References

1. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Proc. CRYPTO 2000*, pages 255–270. Springer-Verlag, 2000. Lecture Notes in Computer Science No. 1880.
2. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. In *Proc. EUROCRYPT 2004*, pages 268–286. Springer-Verlag, 2004. Lecture Notes in Computer Science No. 3027.
3. M. Bellare, H. Shi, and C. Zhang. Foundations of group signature: The case of dynamic groups. To appear in CT-RSA 2005, 2005.
4. T. Beth. Efficient zero-knowledged identification scheme for smart cards. In *Proc. EUROCRYPT 88*, pages 77–86. Springer-Verlag, 1988. Lecture Notes in Computer Science No. 330.
5. D. Boneh and X. Boyen. Short signatures without random oracles. In *Proc. EUROCRYPT 2004*, pages 56–73. Springer-Verlag, 2004. Lecture Notes in Computer Science No. 3027.
6. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Proc. CRYPTO 2004*, pages 41–55. Springer-Verlag, 2004. Lecture Notes in Computer Science No. 3152.
7. D. Boneh and M. Franklin. Identity-based encryption from the weil paring. In *Proc. CRYPTO 2001*, pages 213–229. Springer-Verlag, 2001. Lecture Notes in Computer Science No. 2139.
8. E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. Cryptology ePrint Archive, Report 2004/205, 2004. http://eprint.iacr.org/.
9. J. Camenisch and I. Damgård. Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In *Proc. ASIACRYPT 2000*, pages 331–345. Springer-Verlag, 2000. Lecture Notes in Computer Science No. 1976.
10. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In *Proc. CRYPTO 97*, pages 410–424. Springer-Verlag, 1997. Lecture Notes in Computer Science No. 1294.
11. C. Castelluccia. How to convert any ID-based signature schemes into a group signature scheme. Cryptology ePrint Archive, Report 2002/116, 2002. http://eprint.iacr.org/.
12. J.C. Cha and J.H. Cheon. An identity-based signature from gap diffie-hellman groups. In *Practice and Theory in Public Key Cryptography – PKC'2003*, pages 18–30. Springer-Verlag, 2003. Lecture Notes in Computer Science No. 2567.
13. D. Chaum and E. van Heyst. Group signatures. In *Proc. EUROCRYPT 91*, pages 257–265. Springer-Verlag, 1991. Lecture Notes in Computer Science No. 547.

14. X. Chen, F. Zhang, and K. Kim. A new ID-based group signature scheme from bilinear pairings. Cryptology ePrint Archive, Report 2002/184, 2002. http://eprint.iacr.org/.
15. Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous identification in ad hoc groups. In *Proc. EUROCRYPT 2004*, pages 609–626. Springer-Verlag, 2004. Lecture Notes in Computer Science No. 3027.
16. F. Hess. Efficient identity based signature schemes based on pairing. In *Selected Area in Cryptography, SAC2002*, pages 310–324. Springer-Verlag, 2003.
17. M. Joye. On the difficulty coalition-resistance in group signature schemes (ii). *Technique Report*, LCIS-99-6B, 1999.
18. M. Joye, S. Kim, and N. Lee. Cryptanalysis of two group signature schemes. In *Information Security 1999*, pages 271–275. Springer-Verlag, 1999. Lecture Notes in Computer Science No. 1729.
19. W. Mao and C.H. Lim. Cryptanalysis in prime order subgroup of Z_n. In *Proc. ASIACRYPT 98*, pages 214–226. Springer-Verlag, 1998. Lecture Notes in Computer Science No. 1514.
20. S. Mitsunari, R. Sakai, and M. Kasahara. A new traitor tracing. IEICE Trans. Fundamentals, E85-A(2):481-4, 2002.
21. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *Proc. CRYPTO 92*, pages 31–53. Springer-Verlag, 1992. Lecture Notes in Computer Science No. 740.
22. S. Park, S. Kim, and D. Won. ID-based group signature. In *Electronics Letters, 1997, 33(15)*, pages 1616–1617, 1997.
23. R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *Proc. ASIACRYPT 2001*, pages 552–565. Springer-Verlag, 2001. Lecture Notes in Computer Science No. 2248.
24. R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *Proc. of SCIS 2000*, 2000.
25. A. Shamir. Identity-based cryptosystems and signature schemes. In *Proc. CRYPTO 84*, pages 47–53. Springer-Verlag, 1984. Lecture Notes in Computer Science No. 196.
26. Y. Tseng and J. Jan. A novel ID-based group signature. In *International computer symposium, workshop on cryptology and information security*, pages 159–164, 1998.
27. Victor K. Wei. Tight reductions among strong diffie-hellman assumptions. Cryptology ePrint Archive, Report 2005/057, 2005. http://eprint.iacr.org/.
28. F. Zhang, R. Safavi-Naini, and W. Susilo. An efficient signature scheme from bilinear pairings and its applications. In *Proc. PKC'2004*, pages 277–290. Springer-Verlag, 2004. Lecture Notes in Computer Science No. 2947.