RAIRO-Theor. Inf. Appl. 50 (2016) 193–226
DOI: 10.1051/ita/2016021

Available online at:
www.rairo-ita.org

TRACEABLE IDENTITY-BASED GROUP SIGNATURE

Ke Gu[1,2], Lihao Yang[1], Yong Wang[2] and Sheng Wen[3]

Abstract. Group signature is a useful cryptographic primitive, which makes every group member sign messages on behalf of a group they belong to. Namely group signature allows that group member anonymously signs any message without revealing his/her specific identity. However, group signature may make the signers abuse their signing rights if there are no measures of keeping them from abusing signing rights in the group signature schemes. So, group manager must be able to trace (or reveal) the identity of the signer by the signature when the result of the signature needs to be arbitrated, and some revoked group members must fully lose their capability of signing a message on behalf of the group they belong to. A practical model meeting the requirement is verifier-local revocation, which supports the revocation of group member. In this model, the verifiers receive the group member revocation messages from the trusted authority when the relevant signatures need to be verified. With the rapid development of identity-based cryptography, several identity-based group signature (IBGS) schemes have been proposed. Compared with group signature based on public key cryptography, IBGS can simplify key management and be used for more applications. Although some identity-based group signature schemes have been proposed, few identity-based group signature schemes are constructed in the standard model and focus on the traceability of signature. In this paper, we present a fully traceable (and verifier-local revocation) identity-based group signature (TIBGS) scheme, which has a security reduction to the computational Diffie–Hellman (CDH) assumption. Also, we give a formal security model for traceable identity-based group signature and prove that the proposed scheme has the properties of traceability and anonymity.

Mathematics Subject Classification. 94A60.

1. Introduction

1.1. Background

Group signature [17] allows group member (signer) to hide his identifying information to a group when group member signs messages, thus group signature only reveals the fact that a message was signed by possible one of group members (a list of possible signers). Additionally, in a practical group signature scheme, the group must be constructed by a group manager, who can revoke the anonymity of any signer or identify the real group

Keywords and phrases. Group signature, identity-based cryptography, traceability, security model.

[1] School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, P.R. China. gk4572@163.com

[2] School of Information Science and Engineering, Central South University, Changsha 410083, P.R. China.

[3] School of Information Technology, Deakin University, Melbourne, Australia.

Article published by EDP Sciences

signer. Because a list of possible signers must be constructed to form a group, some intricate problems need to be solved, such as joining the new members and the revocation of group members. Ateniese *et al.* [3] first proposed an efficient and provably coalition-resistant group signature scheme. However, the security of coalition-resistant group signature was not formalized. In [6], Bellare *et al.* summarized the requirements of group signature and showed the security definitions of group signature. Boneh *et al.* [10] proposed a short group signature scheme in the random oracle model.

In public key cryptography, the management of public keys is a critical problem. For example, certificate authority (CA) generates a digital certificate, which assures that public key belongs to the corresponding user. Then, in a group signature scheme based on public key cryptography, a group public key is corresponding to multi-distributing private keys (signing keys), the joining and revocation of group member is an intricate problem [4, 9, 11, 14]. For large group, it is inefficient to update group public key and distributing private keys when a user joins or exits a group. Bresson *et al.* [11] proposed that the signer may prove that his group certificate does not belong to a list of revoked certificates. However, the length of group signature is proportional to the number of revoked group members. Camenisch *et al.* [14] proposed a different way to handle this problem by using accumulators[4]. However, in some pairing-based accumulators [15,26], the size of public keys linearly grows with the maximal number of accumulations.

The method of verifier-local revocation was proposed by Brickell in [12]. Boneh *et al.* [9] gave the formal definitions of verifier-local revocation. In this kind of approaches [13,21,24,33], the verifiers receive the revocation list of group members from the authority (such as private key generator) when a signature needs to be verified, and non-revoked group members do not need to update their distributing private keys. So, the length of signature does not depend on the number of revoked group members in this model, and the verifiers only need to perform an additional computing to test that whether the signature was signed by a revoked group member on the revocation list of group members. Of course, this kind of approaches increase the verification cost being proportional to the size of the revocation list.

In 2009, Nakanishi *et al.* [25] proposed a revocable group signature scheme with constant complexities for signing and verifying. Also, group members do not need to update their distributing private keys. However, the size of public keys linearly grows with the maximal number N of users in their scheme. In 2012, Libert *et al.* [22,23] proposed two group signature schemes based on public key cryptography, which have many useful properties [23]: $\mathrm{O}(\log N)$-size group public keys, revocation lists of size $\mathrm{O}(r)$ ((r) is the number of revoked users), constant membership certificate size, constant signature size and verification time.

Identity-based cryptography is another cryptographic primitive. In identity-based cryptography, a user's public key is obtained from his public identity, such as name, IP address or email address, *etc.* Then, the user's private key is distributed from a private key generator (PKG). The main target of application of identity-based cryptography is to simplify key management and remove public key certificates. In the group signature schemes based on public key cryptography, the proposed schemes suffers from many drawbacks such as verification and revocation of certificates. Obviously, removing public key certificates can simplify the procedure of joining and revocation of group member. So, compared with group signature based on public key cryptography, identity-based group signature can lessen the suffering of joining and revocation of group member. Identity-based group signature allows a group member to sign a message by the identity of a group that he belongs to, and does not reveal the specific identity of the group member, while the group manager can trace the identity of the group member by the signature if the result of the signature needs to be arbitrated. Also, the receiver of the group signature verifies the signature by the identity of the group that the signer belongs to. When a group member leaves the group or joins the group, identity-based group signature revokes or verifies his membership by not dealing with his public key certificate but dealing with his identity. Identity-based group signature can simplify key management and be more easily used for many applications, such as e-voting, distributed systems, grid computing, mobile agent applications, distributed shared object systems, global distribution networks, mobile

[4]An accumulator is a kind of "hash" function mapping a set of values to a short, constant-size string while allowing to efficiently prove that a specific value was accumulated.

communications, and so on. For example, an anonymous e-voting is being done on a BBS-Suppose that a group is discussing an issue on a bulletin board via the Internet and anonymously wishes to vote for the issue on behalf of the group. When a decision is achieved on the group, one of the group can anonymously vote on behalf of the group by identity-based group signature. Obviously, compared with other cryptographic primitives, such as identity-based multi-proxy signature, identity-based group signature can anonymously be used to vote and trace the real signer when the result of the signature needs to be arbitrated.

1.2. Our contributions

In this paper, we present a traceable identity-based group signature scheme in the standard model. Also, we give the formal security models for traceable identity-based group signature. Under our security models, the proposed scheme is proved to have the properties of anonymity and traceability with enough security. In this paper, our contributions are as follows:

- We present a fully traceable (and verifier-local revocation) identity-based group signature scheme in the standard model. No poly-time adversary can produce a valid TIBGS signature on any identities and messages when the adversary may adaptively be permitted to choose identities and messages after executing group-setup oracle, join-user oracle, revoke-user oracle, signature oracle and trace-user oracle.
- We present a framework for TIBGS and show a detailed security model for TIBGS. Compared with the security models of TIBGS [18, 20], we introduce the Libert *et al.*'s model [23] to our security model. In our security model, we consider three situations for the security of TIBGS and further strengthen our security model on identity-based cryptography. Under our security model, the proposed TIBGS scheme is proved to be secure in the standard model, and has a security reduction to the simple standard assumption (computational Diffie–Hellman assumption).
- Compared with other revocable identity-based group signature schemes proposed by [18, 20], the proposed TIBGS scheme has some advantages (the comparisons of the three schemes are given in Appendix A).

1.3. Outline

The rest of this paper is organized as follows. In Section 2, we discuss the related works about IBGS. In Section 3, we review the bilinear pairings and complexity assumptions on which we build. In Section 4, we show a framework for TIBGS. In Section 5, we set up the security models for TIBGS. In Section 6, we propose a traceable identity-based group signature scheme in the standard model under our framework for TIBGS. In Section 7, we analyze the correctness, efficiency and security of the proposed scheme. Finally, we draw our conclusions in Section 8.

2. Related work

Due to the contributions of Boneh *et al.* [7, 8, 27, 29], a rapid development of identity-based cryptography has taken place. Boneh [7] proposed an identity-based encryption scheme in the random oracle model. Waters [29] proposed an efficient identity-based encryption scheme in the standard model. Based on their works, some researchers proposed many identity-based signature schemes in the random oracle model or standard model [5, 16, 19, 27]. Also, with these identity-based signature (IBS) schemes, a lot of variants, such as the identity-based proxy signature (IBPS) schemes [28, 30, 31], the identity-based ring signature schemes [1, 2, 32], the identity-based group signature schemes [18, 20], *etc.*, have also been proposed. In 2012, Au *et al.* [2] proposed a new identity-based event-oriented linkable ring signature scheme with an option as revocable-iff-linked. With this option, if a user generates two linkable ring signatures in the same event, everyone can compute his identity from these two signatures. Presently some identity-based group signature schemes are proposed in the standard model or random oracle model. In 2011, Ibraimi *et al.* [20] proposed an identity-based group signature with membership revocation in the standard model. However, their security model is not enough complete for identity-based group signature, some notions are confused. And their scheme is not fully identity-based group signature scheme, the master key of the system is still constructed on public key cryptography. In 2014, Emura *et al.* [18] proposed an γ-hiding revocable group signature scheme in the random oracle model. Because their scheme introduces the notion of attributes, their scheme is enough complex and inefficient.

3. Preliminaries

3.1. Bilinear maps

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be groups of prime order q and g be a generator of $\mathbb{G}_1$. We say $\mathbb{G}_2$ has an admissible bilinear map, $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ if the following two conditions hold. The map is bilinear; for all a, b, we have $e\left(g^a, g^b\right) = e(g,g)^{a \cdot b}$. The map is non-degenerate; we must have that $e\left(g, g\right) \neq 1$.

3.2. Computational Diffie–Hellman assumption

Definition 3.1 (Computational Diffie–Hellman (CDH) problem)**.** Let $\mathbb{G}_1$ be a group of prime order q and g be a generator of $\mathbb{G}_1$; for all $(g, g^a, g^b) \in \mathbb{G}_1$, with $a, b \in \mathbb{Z}_q$, the CDH problem is to compute $g^{a \cdot b}$.

Definition 3.2. The $(\hbar, \varepsilon)$-CDH assumption holds if no $\hbar$-time algorithm can solve the CDH problem with probability at least ε.

4. A framework for TIBGS

In the section, we present a formal definition of TIBGS. Let $\mathbb{A}$ be universe of possible identities, we set $ID \subseteq \mathbb{A}$ as the identity of user or group.

Definition 4.1 (Traceable Identity-Based Group Signature Scheme)**.** Let **TIBGS** = (*System-Setup, Group-Setup, Join-User, Revoke-User, Sign, Verify, Trace-User*) be a traceable identity-based group signature scheme on $\mathbb{A}$. In **TIBGS**, all algorithms are described as follows:

(1) *System-Setup*: The randomized algorithm run by private key generator (PKG) inputs a security parameter 1^k, and then outputs all system parameters $TIBGK$ and a system private key *spk* on the security parameter 1^k.
(2) *Group-Setup*: The randomized algorithm run by private key generator inputs ($TIBGK$, *spk*, $ID_g \subseteq \mathbb{A}$), and then outputs a group private key sk_{ID_g} to a group manager, where ID_g is a group identity, sk_{ID_g} is a group private key on the management of the group manager.
(3) *Join-User*: The randomized algorithm run by the group manager inputs ($TIBGK$, sk_{ID_g}, $ID_i \subseteq \mathbb{A}$), and then outputs a member private key sk_{ID_i} to a group member, where sk_{ID_i} is the member private key of the group member, ID_i is the corresponding identity and $i \in \{1, 2, \ldots, n\}$ ($n \in \mathbb{N}$ is a maximal number of group members).
(4) *Revoke-User*: The randomized algorithm run by the group manager inputs ($TIBGK$, sk_{ID_g}, $ID_i \subseteq \mathbb{A}$, RL^t_{ID}), and then outputs an updated revocation list RL^{t+1}_{ID}, where ID_i is the corresponding identity of the revoked user, $RL^t_{ID} = \{\ldots(ID_j, \Re_{ID_j})\ldots\}$ is a revocation list in the duration t (ID_j is the corresponding identity of the revoked user and $\Re_{ID_j}$ is a credential on the corresponding identity).
(5) *Sign*: The randomized algorithm is a standard traceable identity-based group signature algorithm. Signer needs to sign a message $\mathfrak{M} \in \{0,1\}^*$. The algorithm run by a group member inputs ($TIBGK$, sk_{ID_i}, $\mathfrak{M}$), and then outputs a signature σ, where $\sigma \in \{0,1\}^* \cup \{\perp\}$, sk_{ID_i} is the member private key of the group member and ID_i is the corresponding identity with $i \in \{1, 2, \ldots, n\}$.

(6) *Verify*: The signature receivers verify a standard traceable identity-based group signature σ. The deterministic algorithm run by a signature verifier inputs ($TIBGK$, $\mathfrak{M}$, ID_g, σ, RL^t_{ID}), and then outputs the boolean value, *accept* or *reject*.
(7) *Trace-User*: The group manager traces a real group member (signer) on the traceable identity-based group signature σ. The deterministic algorithm run by the group manager inputs ($TIBGK$, $\mathfrak{M}$, sk_{ID_g}, σ, RL^t_{ID}), and then outputs the identity of the real signer or $\perp$.

The *correctness* of **TIBGS** requires that for any $(TIBGK, spk) \leftarrow$ *System-Setup*(1^k), $sk_{ID_g} \leftarrow$ *Group-Setup*$(TIBGK,\ spk,\ ID_g \subseteq \mathbb{A})$, $sk_{ID_i} \leftarrow$ *Join-User*$(TIBGK,\ sk_{ID_g},\ ID_i \subseteq \mathbb{A})$ for all i with $i \in \{1, 2, \ldots, n\}$, $\mathfrak{M} \in \{0,1\}^*$, then

$$\mathbf{Pr}[\mathit{Verify}(TIBGK, \mathfrak{M}, ID_g, \mathit{Sign}(TIBGK, sk_{ID_i}, \mathfrak{M}), RL^t_{ID}) = 1] = 1.$$

The *traceability* of **TIBGS** requires that for any $(TIBGK, spk)$ $\leftarrow$*System-Setup*(1^k), $sk_{ID_g} \leftarrow$ *Group-Setup*$(TIBGK,\ spk,\ ID_g \subseteq \mathbb{A})$, $sk_{ID_i} \leftarrow$ *Join-User*$(TIBGK,\ sk_{ID_g},\ ID_i \subseteq \mathbb{A})$ for all i with $i \in \{1, 2, \ldots, n\}$, $\mathfrak{M} \in \{0,1\}^*$, then

$$\mathbf{Pr}[\mathit{Trace\text{-}User}(TIBGK, \mathfrak{M}, sk_{ID_g}, \mathit{Sign}(TIBGK, sk_{ID_i}, \mathfrak{M}), RL^t_{ID}) = ID_i] = 1,$$

where the identity ID_i belongs to the group named by the identity ID_g.

5. Security model

According to [20, 23], we consider that a fully secure TIBGS scheme must meet the following three security requirements:

(1) *Unforgeability*: A valid TIBGS signature must be signed by a valid group member (signer). Therefore, no poly-time adversary can produce a valid TIBGS signature on any identities and messages when the adversary may adaptively be permitted to choose identities and messages after executing group setup oracle, joining user oracle, revoking user oracle, signature oracle and tracing user oracle.
(2) *Anonymity*: A valid TIBGS signature can only reveal that one group identity possessed by a group manager satisfies the signature. It means a valid TIBGS signature can hide the identifying information of real signer to one group.
(3) *Traceability*: In some situations, a valid TIBGS signature needs to reveal the identity of real signer from one group. It means a valid TIBGS signature can trace a real signer. Then we split the requirement to the following two small security notions[5] [23]:
    (a) The first one is called security against *misidentification attacks*, which requires that even if the adversary can introduce (or corrupt) and revoke any user, a valid TIBGS signature can not reveal the identifying information outside the set of the identities of unrevoked adversarially-controlled users.
    (b) The second one is called security against *framing attacks*, which requires that an honest user is only responsible for the messages that he signed, namely there is no situation that a valid TIBGS signature can reveal the identity of a real group member (signer) but this signer did not sign this signature.

Based on the above three situations, we propose a complete security model for traceable identity-based group signature. Typically, in a security model, security proof is such a process: we first set a computational assumption (problem) not solved under the current computer processing capacity, then we need to illustrate that the ability of the adversary breaking a proposed scheme within a certain time and probability is equal to that of the adversary breaking the unsolved computational problem through the interaction between the adversary and the oracles (algorithms). Therefore, because the setting of the computational problem is impossible to be

[5]The two security notions are more detailedly expanded from the correctness of traceability.

solved under the current computer processing capacity, the adversary does not have the ability to break the proposed scheme, where we call the conversion method of the ability of the adversary as reduction. To make our security model easier to understand, we construct several algorithms interacting with adversary, which may make attack experiments to the traceable identity-based group signature schemes in the above three situations. In our security model, we maximize adversary's advantage, and assume that all attacking conditions needed by adversary hold and adversary may forge signatures after limitedly querying oracles in the above three situations.

In our security model, we assume there are $n+1$ users in a traceable identity-based group signature scheme ($n \in \mathbb{N}$ is a maximal number of group members), and at least one user u^* of $n+1$ users is not corrupted by adversary. And we maximize adversary's advantage, where adversary can get all useful information except for the member private key of u^*[6].

All symbols and parameters are defined as follows in the algorithms:

(1) U^a is a set of users that were registered by an adversary in this game, where the user $u_i^a \in U^a$ with $i \in \{1, 2, \ldots\}$, ID_i^a is the identity of the user u_i^a.
(2) U^b is a set of honest users when an adversary acts a dishonest group manager in this game, where the user $u_i^b \in U^b$ with $i \in \{1, 2, \ldots\}$, ID_i^b is the identity of the user u_i^b.
(3) k is a secure parameter, $\mathcal{A}$ represents an adversary.

Definition 5.1 (Unforgeability of a Traceable Identity-Based Group Signature Scheme)**.** Let **TIBGS** = (*System-Setup, Group-Setup, Join-User, Revoke-User, Sign, Verify, Trace-User*) be a traceable identity-based group signature scheme on $\mathbb{A}$, where $\mathbb{A}$ is the universe of possible identities. Additionally, we set that k is a secure parameter, and $\mathbf{Pr}(\mathcal{B}_{U\_TIBGS}(k,\mathcal{A})=1)$ is the probability that the algorithm $\mathcal{B}_{U\_TIBGS}$ returns 1. Then the advantage that the adversary $\mathcal{A}$ breaks **TIBGS** is defined as follows:

$$\mathrm{Adv}_{TIBGS}^{u\_tibgs-uf}(k, q_g, q_j, q_s, \hbar) = \mathbf{Pr}(\mathcal{B}_{u\_tibgs}(k, \mathcal{A}) = 1),$$

where q_g is the maximal number of "Group-Setup" oracle queries, q_j is the maximal number of "Join-User" oracle queries, q_s is the maximal number of "Sign" oracle queries and $\hbar$ is the running time of $\mathcal{B}$. If the advantage that the adversary breaks **TIBGS** is negligible, then the scheme **TIBGS** is secure.

According to Definition 5.1, the algorithm $\mathcal{B}_{U\_TIBGS}$ is described as follows:

1. *Setup*: Running *System-Setup*, $(TIBGK, spk) \leftarrow$*System-Setup*(1^k), and then $TIBGK$ is passed to $\mathcal{A}$.

2. *Queries*: $\mathcal{A}$ makes queries to the following oracles for polynomially many times:

- *Group-Setup*(): Given the public parameters $TIBGK$ and the identity ID_g of the group, the oracle returns a group private key sk_{ID_g} to $\mathcal{A}$.
- *Join-User*(): Given the public parameters $TIBGK$, the group private key sk_{ID_g} (or the identity ID_g) and the identity ID_i of the group member, the oracle returns a member private key sk_{ID_i} to $\mathcal{A}$, where sk_{ID_g} is a group private key on the identity ID_g of the group.
- *Sign*(): Given the public parameters $TIBGK$, the member private key sk_{ID_i} (or the identity ID_i) and the message $\mathfrak{M}$, the oracle returns a signature σ to $\mathcal{A}$, where $\sigma \in \{0,1\}^* \cup \{\perp\}$, sk_{ID_i} is the member private key of the group member and ID_i is the corresponding identity.

3. *Forgery*: $\mathcal{A}$ outputs its forgery, ($\mathfrak{M}^*$, σ^*) for ID_g^* and $RL_{ID_g^*}^t$, where the identity ID_g^* and the revocation list $RL_{ID_g^*}^t$ are arbitrary forgeries generated by $\mathcal{A}$. It succeeds if

(a) $1 \leftarrow$*Verify*$(TIBGK,\ \mathfrak{M}^*,\ ID_g^*,\ \sigma^*,\ RL_{ID_g^*}^t)$;
(b) $\mathcal{A}$ did not query *Group-Setup* on input ID_g^*, did not query *Join-User* on inputs $sk_{ID_g^*}$ and ID^*, and did not query *Sign* on inputs sk_{ID^*} and $\mathfrak{M}^*$ where the identity ID^* of sk_{ID^*} belongs to the group named by the identity ID_g^*.

[6] u^* is used to play a challenger which can interact with simulator and adversary.

Definition 5.2 (Traceability of a Traceable Identity-Based Group Signature Scheme)**.** Let **TIBGS**=(*System-Setup, Group-Setup, Join-User, Revoke-User, Sign, Verify, Trace-User*) be a traceable identity-based group signature scheme, which meets the requirement of unforgeability. **TIBGS** is traceable if the following conditions can be satisfied:

(1) For all valid generated $(TIBGK, spk) \leftarrow$ *System-Setup*(1^k), $sk_{ID_g} \leftarrow$ *Group-Setup*$(TIBGK,\ spk,\ ID_g)$, sk_{ID_i} $\leftarrow$*Join-User*$(TIBGK,\ sk_{ID_g},\ ID_i)$ with $i \in \{0,1\}$, then σ_0 =*Sign*$(TIBGK,\ sk_{ID_0},\ \mathfrak{M})$ and σ_1 = *Sign*$(TIBGK,\ sk_{ID_1},\ \mathfrak{M})$, the outputs of *Trace-User*$(TIBGK,\ \mathfrak{M},\ sk_{ID_g},\ \sigma_0,\ RL^t_{ID})$ and *Trace-User*$(TIBGK,\ \mathfrak{M},\ sk_{ID_g},\ \sigma_1,\ RL^t_{ID})$ are distinguishable in polynomially many times.

(2) We set that k is a secure parameter, and $\mathbf{Pr}(\mathcal{B}_{TM\_TIBGS}(k,\mathcal{A})=1)$ is the probability that the algorithm $\mathcal{B}_{TM\_TIBGS}$ returns 1, and that $\mathbf{Pr}(\mathcal{B}_{TF\_TIBGS}(k,\mathcal{A})=1)$ is the probability that the algorithm $\mathcal{B}_{TF\_TIBGS}$ returns 1. Then the advantage that the adversary $\mathcal{A}$ breaks **TIBGS** is defined as follows:

$$\mathrm{Adv}^{t\_tibgs-mf}_{TIBGS}(k, q_g, q_j, q_r, q_s, \hbar) = \mathbf{Pr}(\mathcal{B}_{tm\_tibgs}(k, \mathcal{A}) = 1) \parallel \mathbf{Pr}(\mathcal{B}_{tf\_tibgs}(k, \mathcal{A}) = 1),$$

where q_g is the maximal number of "Group-Setup" oracle queries, q_j is the maximal number of "Join-User" oracle queries, q_r is the maximal number of "Revoke-User" oracle queries, q_s is the maximal number of "Sign" oracle queries and $\hbar$ is the running time of $\mathcal{B}$. If the advantage that the adversary breaks **TIBGS** is negligible, then the scheme **TIBGS** is secure.

According to Definition 5.2, the algorithm $\mathcal{B}_{TM\_TIBGS}$ is described as follows:

1. *Setup*: Running *System-Setup*, $(TIBGK, spk) \leftarrow$*System-Setup*(1^k), and then $TIBGK$ is passed to $\mathcal{A}$.

2. *Queries*: $\mathcal{A}$ makes queries to the following oracles for polynomially many times:

- *Join-User*(): Given the public parameters $TIBGK$, the group private key sk_{ID_g} (or the identity ID_g) and the identity $ID_{u_i^a}$ of the group member, the oracle returns a member private key $sk_{ID_{u_i^a}}$ to $\mathcal{A}$, where sk_{ID_g} is a group private key on the identity ID_g of the group and the user (group member) u_i^a is added to the set U^a.
- *Revoke-User*(): Given the public parameters $TIBGK$, the group private key sk_{ID_g} (or the identity ID_g), the identity $ID_{u_i^a}$ of the revoked group member and the revocation list RL^t_{ID} of the last duration t, the oracle returns an updated revocation list RL^{t+1}_{ID}.
- *Sign*(): Given the public parameters $TIBGK$, the member private key $sk_{ID_{u_i^a}}$ (or the identity $ID_{u_i^a}$) and the message $\mathfrak{M}$, the oracle returns a signature σ to $\mathcal{A}$, where $\sigma \in \{0,1\}^* \cup \{\perp\}$, $sk_{ID_{u_i^a}}$ is the member private key of the group member, $ID_{u_i^a}$ is the corresponding identity, and the user u_i^a is added to the set U^a if $u_i^a \notin U^a$.

3. *Forgery*: $\mathcal{A}$ outputs its forgery, $(\mathfrak{M}^*,\ \sigma^*)$ for ID_g^* and $RL^t_{ID_g^*}$, where the identity ID_g^* and the revocation list $RL^t_{ID_g^*}$ are arbitrary forgeries generated by $\mathcal{A}$. It succeeds if

(a) $1 \leftarrow$*Verify*$(TIBGK,\ \mathfrak{M}^*,\ ID_g^*,\ \sigma^*,\ RL^t_{ID_g^*})$;

(b) $\mathcal{A}$ did not query *Join-User* on inputs $sk_{ID_g^*}$ and ID^*, did not query *Revoke-User* on inputs $sk_{ID_g^*}$, ID^* and $RL^{t-1}_{ID_g^*}$, and did not query *Sign* on inputs sk_{ID^*} and $\mathfrak{M}^*$, where the identity ID^* of sk_{ID^*} belongs to the group named by the identity ID_g^* and $ID^* \notin U^a \setminus RL^t_{ID_g^*}$;

(c) $ID^* \leftarrow$*Trace-User*$(TIBGK,\ \mathfrak{M}^*,\ sk_{ID_g^*},\ \sigma^*,\ RL^t_{ID_g^*})$.

And then the algorithm $\mathcal{B}_{TF\_TIBGS}$ is described as follows:

1. *Setup*: Running *System-Setup*, $(TIBGK, spk) \leftarrow$*System-Setup*(1^k), and then $TIBGK$ is passed to $\mathcal{A}$.

2. *Queries*: $\mathcal{A}$ makes queries to the following oracles for polynomially many times:

- *Group-Setup*(): Given the public parameters $TIBGK$ and the identity ID_g of the group, the oracle returns a group private key sk_{ID_g} to $\mathcal{A}$.

- *Join-User*(): Given the public parameters $TIBGK$, the group private key sk_{ID_g} (or the identity ID_g) and the identity $ID_{u_i^b}$ of the group member, the oracle returns a member private key $sk_{ID_{u_i^b}}$ to $\mathcal{A}$, where sk_{ID_g} is a group private key on the identity ID_g of the group and the user (group member) u_i^b is added to the set U^b where $U^b \neq \varnothing$.
- *Revoke-User*(): Given the public parameters $TIBGK$, the group private key sk_{ID_g} (or the identity ID_g), the identity $ID_{u_i^b}$ of the revoked group member and the revocation list RL_{ID}^t of the last duration t, the oracle returns an updated revocation list RL_{ID}^{t+1}.
- *Sign*(): Given the public parameters $TIBGK$, the member private key $sk_{ID_{u_i^b}}$ (or the identity $ID_{u_i^b}$) and the message $\mathfrak{M}$, the oracle returns a signature σ to $\mathcal{A}$, where $\sigma \in \{0,1\}^* \cup \{\perp\}$, $sk_{ID_{u_i^b}}$ is the member private key of the group member, $ID_{u_i^b}$ is the corresponding identity and the user u_i^b is added to the set U^b if $u_i^b \notin U^b$.

3. *Forgery*: $\mathcal{A}$ outputs its forgery, ($\mathfrak{M}^*$, σ^*) for ID_g^* and $RL_{ID_g^*}^t$, where the identity ID_g^* and the revocation list $RL_{ID_g^*}^t$ are arbitrary forgeries generated by $\mathcal{A}$. It succeeds if

(a) $1 \leftarrow$ *Verify*$(TIBGK, \mathfrak{M}^*, ID_g^*, \sigma^*, RL_{ID_g^*}^t)$;

(b) $\mathcal{A}$ did not query *Group-Setup* on input ID_g^*, did not query *Join-User* on inputs $sk_{ID_g^*}$ and ID^*, did not query *Revoke-User* on inputs $sk_{ID_g^*}$, ID^* and $RL_{ID_g^*}^{t-1}$, and did not query *Sign* on inputs sk_{ID^*} and $\mathfrak{M}^*$, where the identity ID^* of sk_{ID^*} belongs to the group named by the identity ID_g^* and $ID^* \in U^b$;

(c) $ID^* \leftarrow$ *Trace-User*$(TIBGK, \mathfrak{M}^*, sk_{ID_g^*}, \sigma^*, RL_{ID_g^*}^t)$.

Definition 5.3 (Anonymity of a Traceable Identity-Based Group Signature Scheme)**.** Let **TIBGS** = (*System-Setup, Group-Setup, Join-User, Revoke-User, Sign, Verify, Trace-User*) be a traceable identity-based group signature scheme. Additionally, we set that k is a secure parameter, and $\mathbf{Pr}(\mathcal{B}_{A\_TIBGS}(k,\mathcal{A})=1)$ is the probability that the algorithm $\mathcal{B}_{A\_TIBGS}$ returns 1. Then the advantage that the adversary $\mathcal{A}$ breaks **TIBGS** is defined as follows:

$$\mathrm{Adv}_{TIBGS}^{a\_tibgs}(k, q_g, q_j, q_r, q_s, \hbar) = \left|\mathbf{Pr}(\mathcal{B}_{a\_tibgs}(k, \mathcal{A}) = 1) - \frac{1}{2}\right|,$$

where q_g is the maximal number of "Group-Setup" oracle queries, q_j is the maximal number of "Join-User" oracle queries, q_r is the maximal number of "Revoke-User" oracle queries, q_s is the maximal number of "Sign" oracle queries and $\hbar$ is the running time of $\mathcal{B}$. If the advantage that the adversary breaks **TIBGS** is negligible, then the scheme **TIBGS** is secure.

According to Definition 5.3, the algorithm $\mathcal{B}_{A\_TIBGS}$ is described as follows:

1. *Setup*: Running *System-Setup*, $(TIBGK, spk) \leftarrow$ *System-Setup*(1^k), and then $TIBGK$ is passed to $\mathcal{A}$.

2. *Queries Phase* 1: $\mathcal{A}$ makes queries to the following oracles for polynomially many times:

- *Group-Setup*(): Given the public parameters $TIBGK$ and the identity ID_g of the group, the oracle returns a group private key sk_{ID_g} to $\mathcal{A}$.
- *Join-User*(): Given the public parameters $TIBGK$, the group private key sk_{ID_g} (or the identity ID_g) and the identity ID_i of the group member, the oracle returns a member private key sk_{ID_i} to $\mathcal{A}$, where sk_{ID_g} is a group private key on the identity ID_g of the group.
- *Revoke-User*(): Given the public parameters $TIBGK$, the group private key sk_{ID_g} (or the identity ID_g), the identity ID_i of the revoked group member and the revocation list RL_{ID}^t of the last duration t, the oracle returns an updated revocation list RL_{ID}^{t+1}.
- *Sign*(): Given the public parameters $TIBGK$, the member private key sk_{ID_i} (or the identity ID_i) and the message $\mathfrak{M}$, the oracle returns a signature σ to $\mathcal{A}$, where $\sigma \in \{0,1\}^* \cup \{\perp\}$, sk_{ID_i} is the member private key of the group member and ID_i is the corresponding identity.

3. *Challenge*: $\mathcal{A}$ sends to the challenger its forgery ($\mathfrak{M}^*$, ID_g^*, $RL_{ID_g^*}^t$) and two group member identities ID_0^* and ID_1^* that belong to the group named by the group identity ID_g^*. The forgery satisfies the following conditions:

(a) $\mathcal{A}$ did not query *Group-Setup* on input ID_g^*;
(b) $\mathcal{A}$ did not query *Join-User* on inputs ID_g^*, ID_0^* (and ID_1^*);
(c) $\mathcal{A}$ did not query *Revoke-User* on inputs ID_g^*, ID_0^* (and ID_1^*) and $RL_{ID_g^*}^{t-1}$.

The challenger picks a random bit $x \in \{0,1\}$, and then runs and outputs $\sigma^* \leftarrow$*Sign*($TIBGK$, $sk_{ID_x^*}$, $\mathfrak{M}^*$) to $\mathcal{A}$.

4. *Queries Phase* 2: $\mathcal{A}$ makes queries to the following oracles for polynomially many times again:

- *Group-Setup*(): Given the public parameters $TIBGK$ and the identity ID_g of the group (where $ID_g \neq ID_g^*$), the oracle returns a group private key sk_{ID_g} to $\mathcal{A}$.
- *Join-User*(): Given the public parameters $TIBGK$, the group private key sk_{ID_g} (or the identity ID_g) and the identity ID_i of the group member (where $sk_{ID_g} \neq sk_{ID_g^*}$ and $ID_i \notin \{ID_0^*, ID_1^*\}$), the oracle returns a member private key sk_{ID_i} to $\mathcal{A}$, where sk_{ID_g} is a group private key on the identity ID_g of the group.
- *Revoke-User*(): Given the public parameters $TIBGK$, the group private key sk_{ID_g} (or the identity ID_g), the identity ID_i of the revoked group member and the revocation list RL_{ID}^t of the last duration t, the oracle returns an updated revocation list RL_{ID}^{t+1} (where $\mathcal{A}$ did not query *Revoke-User* on inputs $sk_{ID_g^*}$, ID_0^* (and ID_1^*)).
- *Sign*(): Given the public parameters $TIBGK$, the member private key sk_{ID_i} (or the identity ID_i) and the message $\mathfrak{M}$, the oracle returns a signature σ to $\mathcal{A}$.

5. *Guess*: $\mathcal{A}$ outputs a bit $x' \in \{0,1\}$ and succeeds if $x' = x$.

6. Traceable Identity-Based Group Signature Scheme

Let **TIBGS**=(*System-Setup*, *Group-Setup*, *Join-User*, *Revoke-User*, *Sign*, *Verify*, *Trace-User*) be a traceable identity-based group signature scheme. In **TIBGS**, all algorithms are described as follows:

(1) *TIBGS.System-Setup*: The algorithm run by the PKG system inputs a security parameter 1^k. Additionally, let $\mathbb{G}_1$ and $\mathbb{G}_2$ be groups of prime order q and g be a generator of $\mathbb{G}_1$, and let $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ denote the bilinear map. The size of the group is determined by the security parameter, and we set $\mathbb{A} \subseteq \mathbb{Z}_q$ as the universe of identities. And one hash function, $H : \{0,1\}^* \to \mathbb{Z}_{1^k \cdot q}$ can be defined and used to generate any integer value in $\mathbb{Z}_{1^k \cdot q}$ (where 1^k represents the corresponding decimal number).
Then the system parameters are generated as follows. The algorithm chooses random $a, b \in \mathbb{Z}_q$, and then sets $g_1 = g^a$ and $g_3 = g^b$. Nine group elements g_2, g_4, ϑ, ψ, μ, τ, ϖ, χ and $\kappa \in \mathbb{G}_1$ are randomly chosen. Finally, the algorithm outputs the public parameters $TIBGK$=($\mathbb{G}_1$, $\mathbb{G}_2$, e, g, g_1, g_2, g_3, g_4, ϑ, ψ, μ, τ, ϖ, χ, κ), where g_2^a is seen as the system private key *spk*.
Additionally, the algorithm run by the PKG system generates user's private key with respect to the identity of user. The algorithm inputs ($TIBGK$, *spk*, $ID \subseteq \mathbb{A}$), where ID is the identity of user. And then the algorithm randomly chooses $r_1 \in \mathbb{Z}_q$, computes $x_0 = g_2^a \cdot \vartheta^{r_1 \cdot H(ID)} \cdot \psi^{r_1}$ and $x_1 = g^{r_1}$. The algorithm outputs a private key $sk_{\{ID\}} = \{x_0, x_1\}$ for user.

Remark 6.1. Every user may verify his private key by the following equation:

$$e(x_0, g) = e(g_1, g_2) \cdot e(\vartheta, x_1^{H(ID)}) \cdot e(\psi, x_1).$$

(2) *TIBGS.Group-Setup*: The algorithm run by private key generator inputs ($TIBGK$, *spk*, ID_g), where ID_g is a group identity. And then the algorithm randomly chooses $r_2 \in \mathbb{Z}_q$, computes $y_0 = g_4^b \cdot \mu^{r_2 \cdot H(ID_g)} \cdot \tau^{r_2}$, $y_1 = g^{r_2}$. The algorithm outputs a group private key $gsk_{\{ID_g\}} = \{y_0, y_1\}$ to the group manager.

(3) *TIBGS.Join-User*: The algorithm run by the group manager inputs ($TIBGK$, $gsk_{\{ID_g\}}$, ID), where ID is the identity of group member (user). And then the algorithm randomly chooses $r_3, r_4 \in \mathbb{Z}_q$, computes

$$v_0 = y_0 \cdot \vartheta^{r_3 \cdot H(ID)} \cdot \psi^{r_3} \cdot \varpi^{r_4} = g_4^b \cdot \mu^{r_2 \cdot H(ID_g)} \cdot \tau^{r_2} \cdot \vartheta^{r_3 \cdot H(ID)} \cdot \psi^{r_3} \cdot \varpi^{r_4},$$
$$v_1 = e(\vartheta^{r_3 \cdot H(ID)} \cdot \psi^{r_3}, g),$$
$$v_2 = g^{r_4},\ v_3 = y_1 = g^{r_2},\ v_4 = g^{r_3}.$$

Remark 6.2. v_4 is used to trace the real signer in a group.

Finally, the algorithm outputs a member private key $usk_{\{ID\}} = \{v_0, v_1, v_2, v_3\}$ to the group member.

(4) *TIBGS.Revoke-User*: The algorithm run by the group manager inputs ($TIBGK$, ID, RL^t), where ID is the corresponding identity of the revoked user. And the algorithm computes

$$T = v_1 \cdot e(\vartheta^{H(ID)} \cdot \psi, x_1) = e(\vartheta^{(r_1+r_3) \cdot H(ID)} \cdot \psi^{r_1+r_3}, g).$$

Finally, the algorithm outputs and adds a tuple $[ID, T, v_2]$ to the revocation list RL^t, and then an updated revocation list RL^{t+1} is published by a secure approach, where v_1 and v_2 belong to the member private key of the revoked user and x_1 belongs to the private key of the revoked user.

Remark 6.3. The group manager may get x_1 from the PKG system or the revoked user when the revoked user was registered to the group. This construction does not break the security of the whole scheme according to the Paterson *et al.*'s signature scheme [27]. However, to make our description simpler, the approach of publishing the revocation list is not described in this paper.

(5) *TIBGS.Sign*: A group member needs to sign a message $\mathfrak{M} \in \{0,1\}^*$. The algorithm run by the group member inputs ($TIBGK$, $usk_{\{ID\}}$, $\mathfrak{M}$), and then randomly chooses $r_5, r_6 \in \mathbb{Z}_q$, computes

$$\begin{aligned}
\sigma_0 &= x_0 \cdot v_0 \cdot \vartheta^{r_5 \cdot H(ID)} \cdot \psi^{r_5} \cdot \varpi^{r_5} \cdot \chi^{r_6 \cdot H(\mathfrak{M})} \cdot \kappa^{r_6} \\
&= g_2^a \cdot g_4^b \cdot \vartheta^{(r_1+r_3+r_5) \cdot H(ID)} \cdot \psi^{r_1+r_3+r_5} \cdot \mu^{r_2 \cdot H(ID_g)} \cdot \tau^{r_2} \cdot \varpi^{r_4+r_5} \cdot \chi^{r_6 \cdot H(\mathfrak{M})} \cdot \kappa^{r_6}, \\
\sigma_1 &= e(\vartheta^{H(ID)} \cdot \psi, x_1) \cdot v_1 \cdot e(\vartheta^{r_5 \cdot H(ID)} \cdot \psi^{r_5}, g) \\
&= e(\vartheta^{H(ID)} \cdot \psi, g^{r_1}) \cdot e(\vartheta^{r_3 \cdot H(ID)} \cdot \psi^{r_3}, g) \cdot e(\vartheta^{r_5 \cdot H(ID)} \cdot \psi^{r_5}, g) \\
&= e(\vartheta^{(r_1+r_3+r_5) \cdot H(ID)} \cdot \psi^{r_1+r_3+r_5}, g), \\
\sigma_2 &= v_2 \cdot g^{r_5} = g^{r_4+r_5}, \\
\sigma_3 &= v_3 = g^{r_2}, \\
\sigma_4 &= g^{r_6}.
\end{aligned}$$

Finally, the algorithm outputs a signature $\Phi = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4\}$.

(6) *TIBGS.Verify*: The signature receivers verify a standard traceable identity-based group signature σ. The algorithm run by a signature verifier inputs ($TIBGK$, $\mathfrak{M}$, ID_g, Φ, RL^t), and then the following steps are finished:
   (a) The algorithm computes the following equation:

$$e(\sigma_0, g) = e(g_1, g_2) \cdot e(g_3, g_4) \cdot \sigma_1 \cdot e(\varpi, \sigma_2) \cdot e(\mu^{H(ID_g)} \cdot \tau, \sigma_3) \cdot e(\chi^{H(\mathfrak{M})} \cdot \kappa, \sigma_4).$$

   If the above equation is correct, then the algorithm runs into the next step, otherwise the algorithm outputs the boolean value *reject*.

(b) The algorithm computes the following equation by the revocation list RL^t:

$$\sigma_1 = e\left(\vartheta^{H(ID)} \cdot \psi, \frac{\sigma_2}{v_2}\right) \cdot T.$$

If the above equation is correct, then the algorithm outputs the boolean value *reject*; otherwise, if the algorithm does not find the correcting equation $\sigma_1 = e(\vartheta^{H(ID)} \cdot \psi, \frac{\sigma_2}{v_2}) \cdot T$ on the revocation list RL^t, then the algorithm outputs the boolean value *accept*.

Remark 6.4. $\sigma_1 = e(\vartheta^{H(ID)} \cdot \psi, \frac{\sigma_2}{v_2}) \cdot T$ can denote whether the group member (signer) has been revoked.

(7) *TIBGS.Trace-User*: The algorithm run by the group manager inputs ($TIBGK$, $\mathfrak{M}$, Φ). For any potential identity ID, the algorithm computes the following equation:

$$e\left(\vartheta^{H(ID)} \cdot \psi,\ x_1 \cdot\ v_4 \cdot \frac{\sigma_2}{v_2}\right) = \frac{e(\sigma_0, g)}{e(g_1, g_2) \cdot e(g_3, g_4) \cdot e(\varpi, \sigma_2) \cdot e(\mu^{H(ID_g)} \cdot \tau, \sigma_3) \cdot e(\chi^{H(\mathfrak{M})} \cdot \kappa, \sigma_4)}.$$

If the above equation is correct, then the algorithm outputs the identity ID of the real signer.

7. Analysis of the proposed scheme

7.1. Correctness

In the proposed scheme, the traceable identity-based group signature is $\Phi = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4\}$, where

$$\begin{aligned}
\sigma_0 &= x_0 \cdot v_0 \cdot \vartheta^{r_5 \cdot H(ID)} \cdot \psi^{r_5} \cdot \varpi^{r_5} \cdot \chi^{r_6 \cdot H(\mathfrak{M})} \cdot \kappa^{r_6} \\
&= g_2^a \cdot g_4^b \cdot \vartheta^{(r_1+r_3+r_5)\cdot H(ID)} \cdot \psi^{r_1+r_3+r_5} \cdot \mu^{r_2 \cdot H(ID_g)} \cdot \tau^{r_2} \cdot \varpi^{r_4+r_5} \cdot \chi^{r_6 \cdot H(\mathfrak{M})} \cdot \kappa^{r_6}, \\
\sigma_1 &= e(\vartheta^{H(ID)} \cdot \psi, x_1) \cdot v_1 \cdot e(\vartheta^{r_5 \cdot H(ID)} \cdot \psi^{r_5}, g) \\
&= e(\vartheta^{H(ID)} \cdot \psi, g^{r_1}) \cdot e(\vartheta^{r_3 \cdot H(ID)} \cdot \psi^{r_3}, g) \cdot e(\vartheta^{r_5 \cdot H(ID)} \cdot \psi^{r_5}, g) \\
&= e(\vartheta^{(r_1+r_3+r_5)\cdot H(ID)} \cdot \psi^{r_1+r_3+r_5}, g), \\
\sigma_2 &= v_2 \cdot g^{r_5} = g^{r_4+r_5}, \\
\sigma_3 &= v_3 = g^{r_2}, \\
\sigma_4 &= g^{r_6}.
\end{aligned}$$

So, Φ may be verified by the following equation:

$$\begin{aligned}
e(\sigma_0, g) &= e(g_2^a \cdot g_4^b \cdot \vartheta^{(r_1+r_3+r_5)\cdot H(ID)} \cdot \psi^{r_1+r_3+r_5} \cdot \mu^{r_2 \cdot H(ID_g)} \cdot \tau^{r_2} \cdot \varpi^{r_4+r_5} \cdot \chi^{r_6 \cdot H(\mathfrak{M})} \cdot \kappa^{r_6}, g) \\
&= e(g_2^a, g) \cdot e(g_4^b, g) \cdot e(\vartheta^{(r_1+r_3+r_5)\cdot H(ID)} \cdot \psi^{r_1+r_3+r_5}, g) \cdot e(\mu^{r_2 \cdot H(ID_g)} \cdot \tau^{r_2}, g) \cdot e(\varpi^{r_4+r_5}, g) \\
&\quad \times e(\chi^{r_6 \cdot H(\mathfrak{M})} \cdot \kappa^{r_6}, g) \\
&= e(g_1, g_2) \cdot e(g_3, g_4) \cdot \sigma_1 \cdot e(\varpi, \sigma_2) \cdot e(\mu^{H(ID_g)} \cdot \tau, \sigma_3) \cdot e(\chi^{H(\mathfrak{M})} \cdot \kappa, \sigma_4).
\end{aligned}$$

7.2. Efficiency

In the proposed scheme, $\Phi = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4\}$, where

$$\begin{aligned}
\sigma_0 &= x_0 \cdot v_0 \cdot \vartheta^{r_5 \cdot H(ID)} \cdot \psi^{r_5} \cdot \varpi^{r_5} \cdot \chi^{r_6 \cdot H(\mathfrak{M})} \cdot \kappa^{r_6}, \\
\sigma_1 &= e(\vartheta^{H(ID)} \cdot \psi, x_1) \cdot v_1 \cdot e(\vartheta^{r_5 \cdot H(ID)} \cdot \psi^{r_5}, g), \\
\sigma_2 &= v_2 \cdot g^{r_5}, \quad \sigma_3 = v_3 = g^{r_2}, \quad \sigma_4 = g^{r_6}.
\end{aligned}$$

Thus, the length of signature is $4 \cdot |\mathbb{G}_1| + |\mathbb{G}_2|$, where $|\mathbb{G}_1|$ is the size of element in $\mathbb{G}_1$ and $|\mathbb{G}_2|$ is the size of element in $\mathbb{G}_2$. Additionally, because $x_0 \cdot v_0 \cdot \vartheta^{r_5 \cdot H(ID)} \cdot \psi^{r_5} \cdot \varpi^{r_5} \cdot \kappa^{r_6}$, χ^{r_6} in $\chi^{r_6 \cdot H(\mathfrak{M})}$, σ_1, σ_2 and σ_4 may be precomputed, and we assume that the time for integer multiplication and hash computation can be ignored, signing a message for a traceable identity-based group signature only needs to compute at most 1 exponentiation in $\mathbb{G}_1$ and 1 multiplication in $\mathbb{G}_1$. Also, the signature receiver needs to verify a traceable identity-based group signature by the following equations:

(1) $e(\sigma_0, g) = e(g_1, g_2) \cdot e(g_3, g_4) \cdot \sigma_1 \cdot e(\varpi, \sigma_2) \cdot e(\mu^{H(ID_g)} \cdot \tau, \sigma_3) \cdot e(\chi^{H(\mathfrak{M})} \cdot \kappa, \sigma_4)$;
(2) $\sigma_1 = e(\vartheta^{H(ID)} \cdot \psi, \frac{\sigma_2}{v_2}) \cdot T$.

Because the value $e(g_1, g_2) \cdot e(g_3, g_4)$ can be precomputed and cached, verification requires $L + 4$ pairing computations, $L + 2$ exponentiations in $\mathbb{G}_1$, $L + 3$ multiplications in $\mathbb{G}_1$ and $L + 4$ multiplications in $\mathbb{G}_2$, where L is the number of the revoked users in the revocation list RL^t[7].

In this paper, we compare the proposed scheme (the scheme of Sect. 6) with the revocable identity-based group signature scheme proposed by Ibraimi *et al.* [20] and the γ-hiding revocable group signature scheme proposed by Emura *et al.* [18]. In Appendix A, we show the comparisons of the three schemes.

7.3. Security

In the section, we show the proposed scheme (the scheme of Sect. 6) has a security reduction to the CDH assumption and the TIBGS unforgeability under the adaptive chosen message and identity attacks, and has the TIBGS traceability and the TIBGS anonymity. Our proofs for the following theorems are based on the security models of Section 5 (we defer the proofs to Appendix B).

Theorem 7.1. *The scheme of Section 6 is ($\hbar$, ε, q_g, q_j, q_s)-unforgeable (according to Def. 5.1), assuming that the ($\hbar'$, ε')-CDH assumption holds in $\mathbb{G}_1$, where:*

$$\begin{aligned}\varepsilon' &= \left(1 - \frac{q_g}{q}\right) \cdot \left(1 - \frac{q_j}{q}\right) \cdot \left(1 - \frac{q_s}{q}\right)^2 \cdot \frac{\varepsilon}{q^3},\\ \hbar' &= \hbar + O(q_g \cdot (5 \cdot C_{exp} + 4 \cdot C_{mul}) + q_j \cdot (10 \cdot C_{exp} + 7 \cdot C_{mul} + 1 \cdot C_{pair})\\ &\quad + q_s \cdot (15 \cdot C_{exp} + 12 \cdot C_{mul} + 1 \cdot C_{pair}))\end{aligned}$$

and q_g is the maximal number of "Group-Setup" oracle queries, q_j is the maximal number of "Join-User" oracle queries, q_s is the maximal number of "Sign" oracle queries, C_{mul} and C_{exp} are respectively the time for a multiplication and an exponentiation in $\mathbb{G}_1$, C_{pair} is the time for a pairing computation.

Theorem 7.2. *The scheme of Section 6 is a traceable TIBGS scheme when it is unforgeable (Thm. 7.1 holds) and satisfies the following conditions (according to Def. 5.2):*

(a) *The outputs of "Trace-User" oracle are distinguishable in polynomially many times.*
(b) *The scheme of Section 6 is ($\hbar''$, ε'', q_g, q_j, q_r, q_s)-secure, assuming that the ($\hbar'$, ε')-CDH assumption holds in $\mathbb{G}_1$, where:*

$$\begin{aligned}\varepsilon'' &= \left[\frac{\varepsilon' \cdot q^3}{(1-\frac{q_j}{q}) \cdot (1-\frac{q_r}{q}) \cdot (1-\frac{q_s}{q})^2}\right] \Bigg\| \left[\frac{\varepsilon' \cdot q^3}{(1-\frac{q_g}{q}) \cdot (1-\frac{q_j}{q}) \cdot (1-\frac{q_r}{q}) \cdot (1-\frac{q_s}{q})^2}\right],\\ \hbar'' &= \mathrm{MAX}\{\hbar' - O(q_j \cdot (10 \cdot C_{exp} + 7 \cdot C_{mul_1} + 1 \cdot C_{pair}) + q_r \cdot (6 \cdot C_{exp} + 3 \cdot C_{mul_1} + 2 \cdot C_{pair} + C_{mul_2})\\ &\quad + q_s \cdot (15 \cdot C_{exp} + 12 \cdot C_{mul_1} + 1 \cdot C_{pair})),\ \ \hbar' - O(q_g \cdot (5 \cdot C_{exp} + 4 \cdot C_{mul_1})\\ &\quad + q_j \cdot (10 \cdot C_{exp} + 7 \cdot C_{mul_1} + 1 \cdot C_{pair}) + q_r \cdot (6 \cdot C_{exp} + 3 \cdot C_{mul_1} + 2 \cdot C_{pair} + C_{mul_2})\\ &\quad + q_s \cdot (15 \cdot C_{exp} + 12 \cdot C_{mul_1} + 1 \cdot C_{pair}))\},\end{aligned}$$

[7] We only consider the bad thing that the revoked user is the last one in the revocation list when verification starts from the first one to the last one.

and q_g is the maximal number of "Group-Setup" oracle queries, q_j is the maximal number of "Join-User" oracle queries, q_r is the maximal number of "Revoke-User" oracle queries, q_s is the maximal number of "Sign" oracle queries, C_{mul_1} and C_{exp} are respectively the time for a multiplication and an exponentiation in $\mathbb{G}_1$, C_{pair} is the time for a pairing computation and C_{mul_2} is the time for a multiplication in $\mathbb{G}_2$.

Theorem 7.3. *The scheme of Section 6 is ($\hbar$, ε, q_g, q_j, q_r, q_s)-anonymous (according to Def. 5.3), assuming that the ($\hbar'$, ε')-CDH assumption holds in $\mathbb{G}_1$, where:*

$$\varepsilon' = \left(1-\frac{q_{g_1}}{q}\right)\cdot\left(1-\frac{q_{j_1}}{q}\right)\cdot\left(1-\frac{q_{r_1}}{q}\right)\cdot\left(1-\frac{q_{s_1}}{q}\right)^2\cdot\left(1-\frac{q_{g_2}}{q}\right)\cdot\left(1-\frac{q_{j_2}}{q}\right)\cdot\left(1-\frac{q_{r_2}}{q}\right)\cdot\left(1-\frac{q_{s_2}}{q}\right)^2\cdot\frac{\varepsilon-\frac{1}{2}}{q^3},$$
$$\begin{aligned}\hbar' = \hbar + O((q_{g_1}+q_{g_2})\cdot(5\cdot C_{exp}+4\cdot C_{mul_1}) + (q_{j_1}+q_{j_2})\cdot(10\cdot C_{exp}+7\cdot C_{mul_1}+1\cdot C_{pair})\\ + (q_{r_1}+q_{r_2})\cdot(6\cdot C_{exp}+3\cdot C_{mul_1}+2\cdot C_{pair}+C_{mul_2})\\ + (q_{s_1}+q_{s_2})\cdot(15\cdot C_{exp}+12\cdot C_{mul_1}+1\cdot C_{pair})),\end{aligned}$$

q_{g_1} and q_{g_2} are respectively the maximal numbers of "Group-Setup" oracle queries in the Queries Phase 1 and 2, q_{j_1} and q_{j_2} are respectively the maximal numbers of "Join-User" oracle queries in the Queries Phase 1 and 2, q_{r_1} and q_{r_2} are respectively the maximal numbers of "Revoke-User" oracle queries in the Queries Phase 1 and 2, q_{s_1} and q_{s_2} are respectively the maximal numbers of "Sign" oracle queries in the Queries Phase 1 and 2, C_{mul_1} and C_{exp} are respectively the time for a multiplication and an exponentiation in $\mathbb{G}_1$, C_{pair} is the time for a pairing computation and C_{mul_2} is the time for a multiplication in $\mathbb{G}_2$.

8. Conclusions

In this paper, we present a fully traceable (and verifier-local revocation) identity-based group signature scheme, which has a security reduction to the computational Diffie–Hellman (CDH) assumption. Also, we give the formal security models for traceable identity-based group signature. Under our security models, the proposed scheme is proved to have the properties of anonymity and traceability with enough security. Compared with other revocable identity-based group signature schemes proposed by [18, 20], the proposed scheme is efficient. Because the proposed scheme is not enough efficient, the work about TIBGS still needs to be further progressed.

Appendix A. Comparisons of three schemes

Tables A.1–A.3 show the comparisons of the three schemes (the scheme of Sect. 6, the Ibraimi *et al.*'s scheme [20] and the Emura *et al.*'s scheme [18]). Table A.1 shows the signature length comparison of the three schemes. In Table A.1, compared with the Ibraimi *et al.*'s scheme and the Emura *et al.*'s scheme, the signature length of the proposed scheme is the shortest one. Table A.2 shows the performance comparison of the three schemes (where we assume that some computations may be precomputed and the time for integer multiplication and hash computation can be ignored). In Table A.2, compared with the Ibraimi et al.'s scheme, the proposed scheme is efficient on the cost of signing and verification; compared with the Emura et al.'s scheme, although the verification cost of the proposed scheme is more than that of the Emura et al.'s scheme, the signing cost of the proposed scheme is less. Table A.3 shows other comparisons of the three schemes. In Table A.3, our proposed scheme is constructed in the standard model.

Remark A.1. To make the description simpler, we assume that the Emura *et al.*'s scheme is also constructed on symmetric bilinear pairing and some public parameters of the Emura *et al.*'s scheme may be not included in the final signature.

TABLE A.1. Signature Length Comparison of Three Schemes. $|\mathbb{G}_1|$ represents the length of element in $\mathbb{G}_1$, $|\mathbb{Z}_q|$ represents the length of element in $\mathbb{Z}_q$, $|\mathbb{G}_2|$ represents the length of element in $\mathbb{G}_2$.

	The length of signature
Scheme [20]	$8 \cdot |\mathbb{G}_1|$
Scheme [18]	$14 \cdot |\mathbb{G}_1| + 20 \cdot |\mathbb{Z}_q|$
Our scheme	$4 \cdot |\mathbb{G}_1| + |\mathbb{G}_2|$

TABLE A.2. Performance Comparison of Three Schemes. L_m is the length of signed message, L_k is the length of identity, L is the number of the revoked users in the revocation list RL^t, C_{mul_1} and C_{exp} are respectively the time for a multiplication and an exponentiation in $\mathbb{G}_1$, C_{pair} is the time for a pairing computation and C_{mul_2} is the time for a multiplication in $\mathbb{G}_2$.

	Signing	Verification
Scheme [20]	$L_m \cdot C_{exp} + (L_m + 1) \cdot C_{mul_1}$	$(L_m + L_k) \cdot C_{mul_1} + (L_m + L_k + 2) \cdot C_{exp}$ $+9 \cdot C_{pair} + 5 \cdot C_{mul_2}$
Scheme [18]	$(L_m + 4) \cdot C_{exp} + (L_m + 1) \cdot C_{mul_1} + 2 \cdot C_{pair}$	$24 \cdot C_{mul_1} + 54 \cdot C_{exp} + 19 \cdot C_{pair} + 15 \cdot C_{mul_2}$
Our scheme	$C_{exp} + C_{mul_1}$	$(L + 3) \cdot C_{mul_1} + (L + 2) \cdot C_{exp} + (L + 4) \cdot C_{pair}$ $+(L + 4) \cdot C_{mul_2}$

TABLE A.3. Other comparisons of three schemes.

	Model	Assumptions
Scheme [20]	standard model	DLIN (decision Linear) and CDH
Scheme [18]	random oracle model	CDH, DDH (decision Diffie–Hellman), DLIN and SDH (strong Diffie–Hellman)
Our scheme	standard model	CDH

APPENDIX B. SECURITY PROOF

Proof of Theorem 7.1

Proof. Let **TIBGS** be a traceable identity-based group signature scheme of Section 6. Additionally, let $\mathcal{A}$ be an ($\hbar$, ε, q_g, q_j, q_s)-adversary attacking **TIBGS**. From the adversary $\mathcal{A}$, we construct an algorithm $\mathcal{B}$, for (g, g^a, g^b)$\in \mathbb{G}_1$, the algorithm $\mathcal{B}$ is able to use $\mathcal{A}$ to compute $g^{a \cdot b}$. Thus, we assume the algorithm $\mathcal{B}$ can solve the CDH with probability at least ε' and in time at most $\hbar'$, contradicting the ($\hbar'$, ε')-CDH assumption. Such a simulation may be created in the following way:

Setup: The PKG system inputs a security parameter 1^k. Additionally, let $\mathbb{G}_1$ and $\mathbb{G}_2$ be groups of prime order q and g be a generator of $\mathbb{G}_1$, and let $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ denote the bilinear map. The size of the group is determined by the security parameter, and we set $\mathbb{A} \subseteq \mathbb{Z}_q$ as the universe of identities. One hash function, $H : \{0,1\}^* \to \mathbb{Z}_{1^k \cdot q}$ can be defined and used to generate any integer value in $\mathbb{Z}_{1^k \cdot q}$ (where 1^k represents the corresponding decimal number).

Then the system parameters are generated as follows. The algorithm chooses random $x_1, x_2 \in \mathbb{Z}_q$, and then sets $g_1 = g^a$, $g_2 = g^b \cdot g^{-x_1}$, $g_3 = g^b$ and $g_4 = g^a \cdot g^{-x_2}$ ($\mathcal{B}$ doesn't know a and b). Also the algorithm chooses ℓ, ∂, ν, λ, η, α and $\pi \in \mathbb{Z}_q$, and then sets $\vartheta = g_2^{\ell} \cdot g$, $\psi = g^{\partial}$, $\mu = g_4^{\nu} \cdot g$, $\tau = g^{\lambda}$, $\varpi = g^{\eta}$, $\chi = g_2^{\alpha} \cdot g$ and $\kappa = g^{\pi}$. Finally, the system outputs the public parameters $TIBGK = (\mathbb{G}_1, \mathbb{G}_2, e, g, g_1, g_2, g_3, g_4, \vartheta, \psi, \mu, \tau, \varpi, \chi, \kappa)$.

Additionally, because the algorithm $\mathcal{B}$ doesn't know a and b, the algorithm can construct all private keys of users by the following computation: for one user u ($ID \subseteq \mathbb{A}$ is the identity of the user u), the algorithm $\mathcal{B}$

chooses a random $r_1 \in \mathbb{Z}_q$ and computes $x_0 = g_1^{-\frac{1}{\ell}} \cdot \vartheta^{r_1} \cdot g_1^{-\frac{\partial}{\ell}\cdot\frac{1}{H(ID)}} \cdot \psi^{\frac{r_1}{H(ID)}}$, $x_1 = (g_1^{-\frac{1}{\ell}} \cdot g^{r_1})^{\frac{1}{H(ID)}}$, and then outputs a private key $sk_{\{ID\}} = \{x_0, x_1\}$ to $\mathcal{A}$.

Remark B.1. To the correctness of $sk_{\{ID\}}$, $sk_{\{ID\}}$ may be changed as follows:

$$\begin{aligned}
x_0 &= g_1^{-\frac{1}{\ell}} \cdot \vartheta^{r_1} \cdot g_1^{-\frac{\partial}{\ell}\cdot\frac{1}{H(ID)}} \cdot \psi^{\frac{r_1}{H(ID)}} \\
&= g_2^a \cdot g_2^{-a} \cdot g_1^{-\frac{1}{\ell}} \cdot \vartheta^{r_1} \cdot g_1^{-\frac{\partial}{\ell}\cdot\frac{1}{H(ID)}} \cdot \psi^{\frac{r_1}{H(ID)}} \\
&= g_2^a \cdot (g_2^{\ell} \cdot g)^{-\frac{a}{\ell}} \cdot \vartheta^{r_1} \cdot g^{a\cdot(-\frac{\partial}{\ell})\cdot\frac{1}{H(ID)}} \cdot \psi^{\frac{r_1}{H(ID)}} \\
&= g_2^a \cdot \vartheta^{-\frac{a}{\ell}} \cdot \vartheta^{r_1} \cdot \psi^{-\frac{a}{\ell}\cdot\frac{1}{H(ID)}} \cdot \psi^{\frac{r_1}{H(ID)}} \\
&= g_2^a \cdot \vartheta^{r_1-\frac{a}{\ell}} \cdot \psi^{\frac{r_1}{H(ID)}-\frac{a}{\ell}\cdot\frac{1}{H(ID)}} \\
&= g_2^a \cdot \vartheta^{r_1-\frac{a}{\ell}} \cdot \psi^{(r_1-\frac{a}{\ell})\cdot\frac{1}{H(ID)}}, \\
x_1 &= (g_1^{-\frac{1}{\ell}} \cdot g^{r_1})^{\frac{1}{H(ID)}} \\
&= (g^{-\frac{a}{\ell}} \cdot g^{r_1})^{\frac{1}{H(ID)}} \\
&= g^{(r_1-\frac{a}{\ell})\cdot\frac{1}{H(ID)}}.
\end{aligned}$$

Setting $r_1' = (r_1 - \frac{a}{\ell}) \cdot \frac{1}{H(ID)}$, $sk_{\{ID\}} = \{x_0, x_1\} = \{g_2^a \cdot \vartheta^{r_1'\cdot H(ID)} \cdot \psi^{r_1'},\ \ g^{r_1'}\}$ is a valid private key, where we assure that $\ell \cdot H(ID) \neq 0$ **mod** q.

Queries: When running the adversary $\mathcal{A}$, the relevant queries can occur according to Definition 5.1. The algorithm $\mathcal{B}$ answers these in the following way:

- *Group-Setup queries*: Given the public parameters $TIBGK$ and the identity ID_g of the group, the algorithm $\mathcal{B}$ similarly constructs a group private key $gsk_{\{ID_g\}} = \{y_0,\ y_1\} = \{g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu}\cdot\frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}},\ \ (g_3^{-\frac{1}{\nu}} \cdot g^{r_2})^{\frac{1}{H(ID_g)}}\}$ to the adversary $\mathcal{A}$. Setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$, $gsk_{\{ID_g\}} = \{y_0,\ y_1\} = \{g_4^b \cdot \mu^{r_2'\cdot H(ID_g)} \cdot \tau^{r_2'},\ \ g^{r_2'}\}$ is a valid private key, where we assure that $\nu \cdot H(ID_g) \neq 0$ **mod** q.
- *Join-User queries*: Given the public parameters $TIBGK$, the identity ID_g of the group and the identity ID of the group member (user), the algorithm chooses random $r_2, r_3, r_4 \in \mathbb{Z}_q$ and computes

$$\begin{aligned}
v_0 &= g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu}\cdot\frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}} \cdot \vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3} \cdot \varpi^{r_4}, \\
v_1 &= e(\vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3}, g), \\
v_2 &= g^{r_4},\ \ v_3 = (g_3^{-\frac{1}{\nu}} \cdot g^{r_2})^{\frac{1}{H(ID_g)}},\ \ v_4 = g^{r_3}.
\end{aligned}$$

  Finally, the algorithm outputs a member private key $usk_{\{ID\}} = \{v_0, v_1, v_2, v_3, v_4\}$ to the adversary $\mathcal{A}$. Similarly, setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$, $usk_{\{ID\}} = \{v_0, v_1, v_2, v_3, v_4\} = \{g_4^b \cdot \mu^{r_2'\cdot H(ID_g)} \cdot \tau^{r_2'} \cdot \vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3} \cdot \varpi^{r_4},\ \ e(\vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3}, g),\ \ g^{r_4},\ \ g^{r_2'},\ \ g^{r_3}\}$ is a valid private key, where we assure that $\nu \cdot H(ID_g) \neq 0$ **mod** q.

Remark B.2. Where we do not consider the traceability of the real signer, thus v_4 is also passed to the adversary $\mathcal{A}$.

- *Sign queries*: given the public parameters $TIBGK$, the identity ID_g of the group, the identity ID of the group member (user) and the message $\mathfrak{M}$, the algorithm chooses random $r_2, r_3, r_4, r_5 \in \mathbb{Z}_q$ and computes

$$
\begin{aligned}
\sigma_0 &= g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu}\cdot\frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}} \cdot \vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3} \cdot \varpi^{r_4} \cdot g_1^{-\frac{1}{\alpha}} \cdot \chi^{r_5} \cdot g_1^{-\frac{\pi}{\alpha}\cdot\frac{1}{H(\mathfrak{M})}} \cdot \kappa^{\frac{r_5}{H(\mathfrak{M})}},\\
\sigma_1 &= e(\vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3}, g),\\
\sigma_2 &= g^{r_4},\\
\sigma_3 &= (g_3^{-\frac{1}{\nu}} \cdot g^{r_2})^{\frac{1}{H(ID_g)}},\\
\sigma_4 &= (g_1^{-\frac{1}{\alpha}} \cdot g^{r_5})^{\frac{1}{H(\mathfrak{M})}},\\
\sigma_5 &= g^{r_3}.
\end{aligned}
$$

Finally, the algorithm outputs a signature $\Phi = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$ to the adversary $\mathcal{A}$. Similarly, we do not consider the traceability of the real signer and maximize the adversary's advantage, thus σ_5 is also passed to the adversary $\mathcal{A}$.

Remark B.3. To the correctness of Φ, Φ may be changed as follows:

$$
\begin{aligned}
\sigma_0 &= g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu}\cdot\frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}} \cdot \vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3} \cdot \varpi^{r_4} \cdot g_1^{-\frac{1}{\alpha}} \cdot \chi^{r_5} \cdot g_1^{-\frac{\pi}{\alpha}\cdot\frac{1}{H(\mathfrak{M})}} \cdot \kappa^{\frac{r_5}{H(\mathfrak{M})}}\\
&= g_4^{b} \cdot g_4^{-b} \cdot g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu}\cdot\frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}} \cdot \vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3} \cdot \varpi^{r_4} \cdot g_2^{a} \cdot g_2^{-a} \cdot g_1^{-\frac{1}{\alpha}} \cdot \chi^{r_5} \cdot g_1^{-\frac{\pi}{\alpha}\cdot\frac{1}{H(\mathfrak{M})}} \cdot \kappa^{\frac{r_5}{H(\mathfrak{M})}}\\
&= g_4^{b} \cdot (g_4^{\nu} \cdot g)^{-\frac{b}{\nu}} \cdot \mu^{r_2} \cdot g^{b\cdot(-\frac{\lambda}{\nu})\cdot\frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}} \cdot \vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3} \cdot \varpi^{r_4} \cdot g_2^{a} \cdot (g_2^{\alpha} \cdot g)^{-\frac{a}{\alpha}} \cdot \chi^{r_5}\\
&\quad \times g^{a\cdot(-\frac{\pi}{\alpha})\cdot\frac{1}{H(\mathfrak{M})}} \cdot \kappa^{\frac{r_5}{H(\mathfrak{M})}}\\
&= g_4^{b} \cdot \mu^{r_2-\frac{b}{\nu}} \cdot \tau^{b\cdot(-\frac{1}{\nu})\cdot\frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}} \cdot \vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3} \cdot \varpi^{r_4} \cdot g_2^{a} \cdot \chi^{r_5-\frac{a}{\alpha}} \cdot \kappa^{a\cdot(-\frac{1}{\alpha})\cdot\frac{1}{H(\mathfrak{M})}} \cdot \kappa^{\frac{r_5}{H(\mathfrak{M})}}\\
&= g_4^{b} \cdot \mu^{r_2-\frac{b}{\nu}} \cdot \tau^{(r_2-\frac{b}{\nu})\cdot\frac{1}{H(ID_g)}} \cdot \vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3} \cdot \varpi^{r_4} \cdot g_2^{a} \cdot \chi^{r_5-\frac{a}{\alpha}} \cdot \kappa^{(r_5-\frac{a}{\alpha})\cdot\frac{1}{H(\mathfrak{M})}},\\
\sigma_3 &= (g_3^{-\frac{1}{\nu}} \cdot g^{r_2})^{\frac{1}{H(ID_g)}} = g^{(r_2-\frac{b}{\nu})\cdot\frac{1}{H(ID_g)}},\\
\sigma_4 &= (g_1^{-\frac{1}{\alpha}} \cdot g^{r_5})^{\frac{1}{H(\mathfrak{M})}} = g^{(r_5-\frac{a}{\alpha})\cdot\frac{1}{H(\mathfrak{M})}}.
\end{aligned}
$$

Setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$ and $r_5' = (r_5 - \frac{a}{\alpha}) \cdot \frac{1}{H(\mathfrak{M})}$, we may get that

$$
\begin{aligned}
\sigma_0 &= g_4^{b} \cdot \mu^{r_2-\frac{b}{\nu}} \cdot \tau^{(r_2-\frac{b}{\nu})\cdot\frac{1}{H(ID_g)}} \cdot \vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3} \cdot \varpi^{r_4} \cdot g_2^{a} \cdot \chi^{r_5-\frac{a}{\alpha}} \cdot \kappa^{(r_5-\frac{a}{\alpha})\cdot\frac{1}{H(\mathfrak{M})}}\\
&= g_4^{b} \cdot \mu^{r_2'\cdot H(ID_g)} \cdot \tau^{r_2'} \cdot \vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3} \cdot \varpi^{r_4} \cdot g_2^{a} \cdot \chi^{r_5'\cdot H(\mathfrak{M})} \cdot \kappa^{r_5'}\\
&= g_2^{a} \cdot g_4^{b} \cdot \vartheta^{r_3\cdot H(ID)} \cdot \psi^{r_3} \cdot \mu^{r_2'\cdot H(ID_g)} \cdot \tau^{r_2'} \cdot \varpi^{r_4} \cdot \chi^{r_5'\cdot H(\mathfrak{M})} \cdot \kappa^{r_5'},\\
\sigma_3 &= g^{r_2'},\\
\sigma_4 &= g^{r_5'},
\end{aligned}
$$

Thus, $\Phi = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$ is a valid signature, where we assure that $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}) \neq 0 \ \mathbf{mod}\ q$.

Forgery: If the algorithm $\mathcal{B}$ does not abort as a consequence of one of the queries above, the adversary $\mathcal{A}$ will, with probability at least ε, return a message $\mathfrak{M}^*$, and a valid identity-based group signature forgery, $\Phi^* = \{\sigma_0^*, \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*\}$ for the identity ID^* of the group member, the identity ID_g^* of the group and the revocation list $RL^t_{ID^*_g}$, where

$$\begin{aligned}
\sigma^*_0 &= g_2^a \cdot g_4^b \cdot \vartheta^{r^*_2 \cdot H(ID^*)} \cdot \psi^{r^*_2} \cdot \mu^{r^*_3 \cdot H(ID^*_g)} \cdot \tau^{r^*_3} \cdot \varpi^{r^*_4} \cdot \chi^{r^*_5 \cdot H(\mathfrak{M}^*)} \cdot \kappa^{r^*_5},\\
\sigma^*_1 &= e(\vartheta^{r^*_2 \cdot H(ID^*)} \cdot \psi^{r^*_2}, g),\\
\sigma^*_2 &= g^{r^*_4},\\
\sigma^*_3 &= g^{r^*_3},\\
\sigma^*_4 &= g^{r^*_5},\\
\sigma^*_5 &= g^{r^*_2}.
\end{aligned}$$

And $\mathcal{A}$ did not query *Group-Setup* on input ID^*_g, did not query *Join-User* on inputs ID^*_g and ID^*, and did not query *Sign* on inputs ID^*_g, ID^* and $\mathfrak{M}^*$.

If $\ell \cdot H(ID^*) \neq 0 \ \mathbf{mod}\ q$, or $\nu \cdot H(ID^*_g) \neq 0 \ \mathbf{mod}\ q$ or $\alpha \cdot H(\mathfrak{M}^*) \neq 0 \ \mathbf{mod}\ q$, then the algorithm $\mathcal{B}$ will abort.

If $\ell \cdot H(ID^*) = 0 \ \mathbf{mod}\ q$, and $\nu \cdot H(ID^*_g) = 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}^*) = 0 \ \mathbf{mod}\ q$, then the algorithm $\mathcal{B}$ computes and outputs

$$\begin{aligned}
&\sqrt[2]{\frac{\sigma^*_0}{g_1^{-x_1} \cdot g_3^{-x_2} \cdot g^{r^*_2 \cdot H(ID^*)} \cdot g^{r^*_2 \cdot \partial} \cdot g^{r^*_3 \cdot H(ID^*_g)} \cdot g^{r^*_3 \cdot \lambda} \cdot g^{r^*_4 \cdot \eta} \cdot g^{r^*_5 \cdot H(\mathfrak{M}^*)} \cdot g^{r^*_5 \cdot \pi}}}\\
&= \sqrt[2]{\frac{g_2^a \cdot g_4^b \cdot \vartheta^{r^*_2 \cdot H(ID^*)} \cdot \psi^{r^*_2} \cdot \mu^{r^*_3 \cdot H(ID^*_g)} \cdot \tau^{r^*_3} \cdot \varpi^{r^*_4} \cdot \chi^{r^*_5 \cdot H(\mathfrak{M}^*)} \cdot \kappa^{r^*_5}}{g_1^{-x_1} \cdot g_3^{-x_2} \cdot g^{r^*_2 \cdot H(ID^*)} \cdot g^{r^*_2 \cdot \partial} \cdot g^{r^*_3 \cdot H(ID^*_g)} \cdot g^{r^*_3 \cdot \lambda} \cdot g^{r^*_4 \cdot \eta} \cdot g^{r^*_5 \cdot H(\mathfrak{M}^*)} \cdot g^{r^*_5 \cdot \pi}}}\\
&= \sqrt[2]{\frac{(g^b \cdot g^{-x_1})^a \cdot (g^a \cdot g^{-x_2})^b \cdot (g_2^{\ell} \cdot g)^{r^*_2 H(ID^*)} \cdot (g^{\partial})^{r^*_2} \cdot (g_4^{\nu} \cdot g)^{r^*_3 H(ID^*_g)} \cdot (g^{\lambda})^{r^*_3} \cdot (g^{\eta})^{r^*_4} \cdot (g_2^{\alpha} \cdot g)^{r^*_5 H(\mathfrak{M}^*)} \cdot (g^{\pi})^{r^*_5}}{g_1^{-x_1} \cdot g_3^{-x_2} \cdot g^{r^*_2 \cdot H(ID^*)} \cdot g^{r^*_2 \partial} \cdot g^{r^*_3 \cdot H(ID^*_g)} \cdot g^{r^*_3 \lambda} \cdot g^{r^*_4 \cdot \eta} \cdot g^{r^*_5 \cdot H(\mathfrak{M}^*)} \cdot g^{r^*_5 \cdot \pi}}}\\
&= g^{a \cdot b},
\end{aligned}$$

which is the solution to the given CDH problem.

Now, we analyze the probability of the algorithm $\mathcal{B}$ not aborting. For the simulation to complete without aborting, we require that all *Group-Setup* queries will have $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$, all *Join-User* queries will have $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$, and all *Sign* queries will have $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}) \neq 0 \ \mathbf{mod}\ q$, and that $\ell \cdot H(ID^*) = 0 \ \mathbf{mod}\ q$, and $\nu \cdot H(ID^*_g) = 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}^*) = 0 \ \mathbf{mod}\ q$ in forgery. If the algorithm $\mathcal{B}$ does not abort, then the following conditions must hold:

(a) $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$ in *Group-Setup* queries, with $i = 1, 2 \dots q_g$;
(b) $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$ in *Join-User* queries, with $i = 1, 2 \dots q_j$;
(c) $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}_i) \neq 0 \ \mathbf{mod}\ q$ in *Sign* queries, with $i = 1, 2 \dots q_s$;
(d) the algorithm $\mathcal{B}$ does not abort in forgery, namely $\ell \cdot H(ID^*) = 0 \ \mathbf{mod}\ q$, and $\nu \cdot H(ID^*_g) = 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}^*) = 0 \ \mathbf{mod}\ q$.

To make the analysis simpler, we will define the events E_i, F_i, T_i, L_i, R^*, F^*, S^* as

E_i: $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$, with $i = 1, 2 \dots q_g$;
F_i: $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$, with $i = 1, 2 \dots q_j$;
T_i: $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$, with $i = 1, 2 \dots q_s$;
L_i: $\alpha \cdot H(\mathfrak{M}_i) \neq 0 \ \mathbf{mod}\ q$, with $i = 1, 2 \dots q_s$;
R^*: $\ell \cdot H(ID^*) = 0 \ \mathbf{mod}\ q$;
F^*: $\nu \cdot H(ID^*_g) = 0 \ \mathbf{mod}\ q$;
S^*: $\alpha \cdot H(\mathfrak{M}^*) = 0 \ \mathbf{mod}\ q$.

Then the probability of $\mathcal{B}$ not aborting is

$$\Pr(not\_abort) = \Pr\left(\bigcap_{i=1}^{q_g} E_i \wedge \bigcap_{i=1}^{q_j} F_i \wedge \bigcap_{i=1}^{q_s} (T_i \wedge L_i) \wedge R^* \wedge F^* \wedge S^*\right).$$

It is easy to see that the events $\bigcap_{i=1}^{q_g} E_i$, $\bigcap_{i=1}^{q_j} F_i$, $\bigcap_{i=1}^{q_s} T_i$, $\bigcap_{i=1}^{q_s} L_i$, R^*, F^* and S^* are independent. Then we may compute

$$\Pr\left(\bigcap_{i=1}^{q_g} E_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_g} \neg E_i\right) = 1 - q_g \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_g}{q};$$
$$\Pr\left(\bigcap_{i=1}^{q_j} F_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_j} \neg F_i\right) = 1 - q_j \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_j}{q};$$
$$\Pr\left(\bigcap_{i=1}^{q_s} T_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_s} \neg T_i\right) = 1 - q_s \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_s}{q};$$
$$\Pr\left(\bigcap_{i=1}^{q_s} L_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_s} \neg L_i\right) = 1 - q_s \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_s}{q};$$
$$\Pr(R^*) = \frac{1^k}{1^k \cdot q} = \frac{1}{q}; \quad \Pr(F^*) = \frac{1^k}{1^k \cdot q} = \frac{1}{q}; \quad \Pr(S^*) = \frac{1^k}{1^k \cdot q} = \frac{1}{q}\cdot$$

Thus,

$$\begin{aligned}\Pr(not\_abort) &= \Pr\left(\bigcap_{i=1}^{q_g} E_i \wedge \bigcap_{i=1}^{q_j} F_i \wedge \bigcap_{i=1}^{q_s} (T_i \wedge L_i) \wedge R^* \wedge F^* \wedge S^*\right)\\ &= \Pr\left(\bigcap_{i=1}^{q_g} E_i\right) \cdot \Pr\left(\bigcap_{i=1}^{q_j} F_i\right) \cdot \Pr\left(\bigcap_{i=1}^{q_s} T_i\right) \cdot \Pr\left(\bigcap_{i=1}^{q_s} L_i\right) \cdot \Pr(R^*) \cdot \Pr(F^*) \cdot \Pr(S^*)\\ &= \left(1 - \frac{q_g}{q}\right) \cdot \left(1 - \frac{q_j}{q}\right) \cdot \left(1 - \frac{q_s}{q}\right)^2 \cdot \frac{1}{q^3}\cdot\end{aligned}$$

So we can get that $\varepsilon' = (1 - \frac{q_g}{q}) \cdot (1 - \frac{q_j}{q}) \cdot (1 - \frac{q_s}{q})^2 \cdot \frac{\varepsilon}{q^3}$.

If the simulation does not abort, the adversary $\mathcal{A}$ will create a valid signature forgery with probability at least ε. The algorithm $\mathcal{B}$ can then compute $g^{a \cdot b}$ from the forgery as shown above. The time complexity of the algorithm $\mathcal{B}$ is dominated by the time for the exponentiations and multiplications in the queries. We assume that the time for integer addition and integer multiplication, and the time for hash computation can both be ignored, then the time complexity of the algorithm $\mathcal{B}$ is

$$\hbar' = \hbar + O(q_g \cdot (5 \cdot C_{exp} + 4 \cdot C_{mul}) + q_j \cdot (10 \cdot C_{exp} + 7 \cdot C_{mul} + 1 \cdot C_{pair}) + q_s \cdot (15 \cdot C_{exp} + 12 \cdot C_{mul} + 1 \cdot C_{pair})).$$

Thus, Theorem 7.1 follows. □

Proof of Theorem 7.2

Proof. According to Definition 5.2, we need to divide the proof to the following three parts:

a) *Correctness (the outputs of "Trace-User" oracle are distinguishable)*:
From the algorithm *TIBGS.Trace-User*, we may know that

1) $\frac{e(\sigma_0,\ g)}{e(g_1,g_2)\cdot e(g_3,g_4)\cdot e(\varpi,\sigma_2)\cdot e(\mu^{H(ID_g)}\cdot\tau,\sigma_3)\cdot e(\chi^{H(\mathfrak{M})}\cdot\kappa,\sigma_4)}$
$= \frac{e(g_2^a\cdot g_4^b\cdot\vartheta^{(r_1+r_3+r_5)\cdot H(ID)}\cdot\psi^{r_1+r_3+r_5}\cdot\mu^{r_2\cdot H(ID_g)}\cdot\tau^{r_2}\cdot\varpi^{r_4+r_5}\cdot\chi^{r_6\cdot H(\mathfrak{M})}\cdot\kappa^{r_6},\ g)}{e(g_1,g_2)\cdot e(g_3,g_4)\cdot e(\varpi,\sigma_2)\cdot e(\mu^{H(ID_g)}\cdot\tau,\sigma_3)\cdot e(\chi^{H(\mathfrak{M})}\cdot\kappa,\sigma_4)}$
$= e(\vartheta^{(r_1+r_3+r_5)\cdot H(ID)}\cdot\psi^{r_1+r_3+r_5}, g),$

2) $e(\vartheta^{H(ID)}\cdot\psi,\ x_1\cdot\ v_4\cdot\frac{\sigma_2}{v_2})$
$= e(\vartheta^{H(ID)}\cdot\psi,\ g^{r_1}\cdot\ g^{r_3}\cdot\frac{g^{r_4+r_5}}{g^{r_4}})$
$= e(\vartheta^{H(ID)}\cdot\psi,\ g^{r_1+r_3+r_5}).$

So, for any potential identity ID, the algorithm *TIBGS.Trace-User* run by the group manager can verify the identity of a real signer by the following equation:

$$e\left(\vartheta^{H(ID)}\cdot\psi,\ x_1\cdot\ v_4\cdot\frac{\sigma_2}{v_2}\right)=\frac{e(\sigma_0,g)}{e(g_1,g_2)\cdot e(g_3,g_4)\cdot e(\varpi,\sigma_2)\cdot e(\mu^{H(ID_g)}\cdot\tau,\sigma_3)\cdot e(\chi^{H(\mathfrak{M})}\cdot\kappa,\sigma_4)}.$$

b) *Misidentification attacks*:

Let **TIBGS** be a traceable identity-based group signature scheme of Section 6. Additionally, let $\mathcal{A}$ be an ($\hbar$, ε, q_j, q_r, q_s)-adversary attacking **TIBGS**. From the adversary $\mathcal{A}$, we construct an algorithm $\mathcal{B}$, for (g, g^a, g^b)$\in\mathbb{G}_1$, the algorithm $\mathcal{B}$ is able to use $\mathcal{A}$ to compute $g^{a\cdot b}$. Thus, we assume the algorithm $\mathcal{B}$ can solve the CDH with probability at least ε' and in time at most $\hbar'$, contradicting the ($\hbar'$, ε')-CDH assumption. According to the algorithm $\mathcal{B}_{TM\_TIBGS}$ of Definition 5.2, such a simulation may be created in the following way (to avoid the symbol confused, we use u_i^A and U^A to replace u_i^a and U^a of the algorithm $\mathcal{B}_{TM\_TIBGS}$):

Setup: The PKG system inputs a security parameter 1^k. Additionally, let $\mathbb{G}_1$ and $\mathbb{G}_2$ be groups of prime order q and g be a generator of $\mathbb{G}_1$, and let $e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_2$ denote the bilinear map. The size of the group is determined by the security parameter, and we set $\mathbb{A}\subseteq\mathbb{Z}_q$ as the universe of identities. One hash function, $H:\{0,1\}^*\to\mathbb{Z}_{1^k\cdot q}$ can be defined and used to generate any integer value in $\mathbb{Z}_{1^k\cdot q}$ (where 1^k represents the corresponding decimal number).

Then the system parameters are generated as follows. The algorithm chooses random $x_1, x_2\in\mathbb{Z}_q$, and then sets $g_1=g^a$, $g_2=g^b\cdot g^{-x_1}$, $g_3=g^b$ and $g_4=g^a\cdot g^{-x_2}$ ($\mathcal{B}$ doesn't know a and b). Also the algorithm chooses ℓ, ∂, ν, λ, η, α and $\pi\in\mathbb{Z}_q$, and then sets $\vartheta=g_2^{\ell}\cdot g$, $\psi=g^{\partial}$, $\mu=g_4^{\nu}\cdot g$, $\tau=g^{\lambda}$, $\varpi=g^{\eta}$, $\chi=g_2^{\alpha}\cdot g$ and $\kappa=g^{\pi}$. Finally, the system outputs the public parameters $TIBGK$=($\mathbb{G}_1$, $\mathbb{G}_2$, e, g, g_1, g_2, g_3, g_4, ϑ, ψ, μ, τ, ϖ, χ, κ).

Additionally, because the algorithm $\mathcal{B}$ doesn't know a and b, the algorithm can construct all private keys of users by the following computation: for one user u (ID is the identity of the user u), the algorithm $\mathcal{B}$ chooses a random $r_1\in\mathbb{Z}_q$ and computes $x_0=g_1^{-\frac{1}{\ell}}\cdot\vartheta^{r_1}\cdot g_1^{-\frac{\partial}{\ell}\cdot\frac{1}{H(ID)}}\cdot\psi^{\frac{r_1}{H(ID)}}$, $x_1=(g_1^{-\frac{1}{\ell}}\cdot g^{r_1})^{\frac{1}{H(ID)}}$, and then outputs a private key $sk_{\{ID\}}=\{x_0,x_1\}$ to $\mathcal{A}$. Similarly, setting $r_1'=(r_1-\frac{a}{\ell})\cdot\frac{1}{H(ID)}$, $sk_{\{ID\}}=\{x_0,x_1\}=\{g_2^a\cdot\vartheta^{r_1'\cdot H(ID)}\cdot\psi^{r_1'},\ \ g^{r_1'}\}$ is a valid private key, where we assure that $\ell\cdot H(ID)\neq 0$ **mod** q.

Queries: When running the adversary $\mathcal{A}$, the relevant queries can occur according to the algorithm $\mathcal{B}_{TM\_TIBGS}$ of Definition 5.2. The algorithm $\mathcal{B}$ answers these in the following way:

- *Join-User queries*: given the public parameters $TIBGK$, the identity ID_g of the group and the identity $ID_{u_i^A}$ of the group member (the user u_i^A is added to the set U^A), the algorithm chooses random $r_2, r_3, r_4\in\mathbb{Z}_q$ and computes

$$v_0=g_3^{-\frac{1}{\nu}}\cdot\mu^{r_2}\cdot g_3^{-\frac{\lambda}{\nu}\cdot\frac{1}{H(ID_g)}}\cdot\tau^{\frac{r_2}{H(ID_g)}}\cdot\vartheta^{r_3\cdot H(ID_{u_i^A})}\cdot\psi^{r_3}\cdot\varpi^{r_4},$$
$$v_1=e(\vartheta^{r_3\cdot H(ID_{u_i^A})}\cdot\psi^{r_3},g),$$
$$v_2=g^{r_4},\ v_3=(g_3^{-\frac{1}{\nu}}\cdot g^{r_2})^{\frac{1}{H(ID_g)}},\ v_4=g^{r_3}.$$

Finally, the algorithm outputs a member private key $usk_{\{ID_{u_i^A}\}} = \{v_0, v_1, v_2, v_3, v_4\}$ to the adversary $\mathcal{A}$. Similarly, setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$,

$$\begin{aligned} usk_{\{ID_{u_i^A}\}} &= \{v_0, v_1, v_2, v_3, v_4\} \\ &= \{g_4^b \cdot \mu^{r_2' \cdot H(ID_g)} \cdot \tau^{r_2'} \cdot \vartheta^{r_3 \cdot H(ID_{u_i^A})} \cdot \psi^{r_3} \cdot \varpi^{r_4}, \quad e(\vartheta^{r_3 \cdot H(ID_{u_i^A})} \cdot \psi^{r_3}, g), \quad g^{r_4}, \quad g^{r_2'}, \quad g^{r_3}\}. \end{aligned}$$

So, $usk_{\{ID_{u_i^A}\}}$ is a valid private key, where we assure that $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$.

Remark B.4. Where we maximize the adversary's advantage, thus v_4 is also passed to $\mathcal{A}$.

- *Revoke-User*: Given the public parameters $TIBGK$, the identity $ID_{u_i^A}$ of the revoked group member and the revocation list RL_{ID}^t of the last duration t ($RL_{ID}^t = \varnothing$ when $t = 0$), the algorithm chooses random $r_1, r_3, r_4 \in \mathbb{Z}_q$ and computes

$$\begin{aligned} T &= e(\vartheta^{H(ID_{u_i^A})} \cdot \psi, \ (g_1^{-\frac{1}{\ell}} \cdot g^{r_1})^{\frac{1}{H(ID_{u_i^A})}}) \cdot e(\vartheta^{r_3 \cdot H(ID_{u_i^A})} \cdot \psi^{r_3}, g), \\ v_2 &= g^{r_4}. \end{aligned}$$

Finally, the algorithm outputs and adds a tuple $[ID_{u_i^A}, T, v_2]$ to the revocation list RL_{ID}^t, and then an updated revocation list RL_{ID}^{t+1} is published to the adversary $\mathcal{A}$. Similarly, setting $r_1' = (r_1 - \frac{a}{\ell}) \cdot \frac{1}{H(ID_{u_i^A})}$,

$$\begin{aligned} T &= e\left(\vartheta^{H(ID_{u_i^A})} \cdot \psi, \ (g_1^{-\frac{1}{\ell}} \cdot g^{r_1})^{\frac{1}{H(ID_{u_i^A})}}\right) \cdot e\left(\vartheta^{r_3 \cdot H(ID_{u_i^A})} \cdot \psi^{r_3}, g\right) \\ &= e\left(\vartheta^{H(ID_{u_i^A})} \cdot \psi, \ g^{(r_1 - \frac{a}{\ell}) \cdot \frac{1}{H(ID_{u_i^A})}}\right) \cdot e\left(\vartheta^{r_3 \cdot H(ID_{u_i^A})} \cdot \psi^{r_3}, g\right) \\ &= e\left(\vartheta^{H(ID_{u_i^A})} \cdot \psi, g^{r_1'}\right) \cdot e\left(\vartheta^{r_3 \cdot H(ID_{u_i^A})} \cdot \psi^{r_3}, g\right) \\ &= e\left(\vartheta^{(r_1' + r_3) \cdot H(ID_{u_i^A})} \cdot \psi^{(r_1' + r_3)}, g\right), \end{aligned}$$

thus the tuple $[ID_{u_i^A}, T, v_2]$ is a valid data, where we assure that $\ell \cdot H(ID_{u_i^A}) \neq 0 \ \mathbf{mod}\ q$.
- *Sign queries*: Given the public parameters $TIBGK$, the identity ID_g of the group, the identity $ID_{u_i^A}$ of the group member (the user u_i^A is added to the set U^A) and the message $\mathfrak{M}$, the algorithm chooses random $r_2, r_3, r_4, r_5 \in \mathbb{Z}_q$ and computes

$$\begin{aligned} \sigma_0 &= g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu} \cdot \frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}} \cdot \vartheta^{r_3 \cdot H(ID_{u_i^A})} \cdot \psi^{r_3} \cdot \varpi^{r_4} \cdot g_1^{-\frac{1}{\alpha}} \cdot \chi^{r_5} \cdot g_1^{-\frac{\pi}{\alpha} \cdot \frac{1}{H(\mathfrak{M})}} \cdot \kappa^{\frac{r_5}{H(\mathfrak{M})}}, \\ \sigma_1 &= e(\vartheta^{r_3 \cdot H(ID_{u_i^A})} \cdot \psi^{r_3}, g), \\ \sigma_2 &= g^{r_4}, \\ \sigma_3 &= (g_3^{-\frac{1}{\nu}} \cdot g^{r_2})^{\frac{1}{H(ID_g)}}, \\ \sigma_4 &= (g_1^{-\frac{1}{\alpha}} \cdot g^{r_5})^{\frac{1}{H(\mathfrak{M})}}, \\ \sigma_5 &= g^{r_3}. \end{aligned}$$

Finally, the algorithm outputs a signature $\Phi = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$ to the adversary $\mathcal{A}$. Similarly, we maximize the adversary's advantage, thus σ_5 is also passed to the adversary $\mathcal{A}$. Setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$ and $r_5' = (r_5 - \frac{a}{\alpha}) \cdot \frac{1}{H(\mathfrak{M})}$, $\Phi = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$ is a valid signature, where we assure that $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}) \neq 0 \ \mathbf{mod}\ q$.

Forgery: If the algorithm $\mathcal{B}$ does not abort as a consequence of one of the queries above, the adversary $\mathcal{A}$ will, with probability at least ε, return a message $\mathfrak{M}^*$, and a valid identity-based group signature forgery, $\Phi^* = \{\sigma_0^*, \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*\}$ for the identity ID^* of the group member, the identity ID_g^* of the group and the revocation list $RL^t_{ID_g^*}$, where

$$\begin{aligned}
\sigma_0^* &= g_2^a \cdot g_4^b \cdot \vartheta^{r_2^* \cdot H(ID^*)} \cdot \psi^{r_2^*} \cdot \mu^{r_3^* \cdot H(ID_g^*)} \cdot \tau^{r_3^*} \cdot \varpi^{r_4^*} \cdot \chi^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot \kappa^{r_5^*},\\
\sigma_1^* &= e(\vartheta^{r_2^* \cdot H(ID^*)} \cdot \psi^{r_2^*}, g),\\
\sigma_2^* &= g^{r_4^*},\\
\sigma_3^* &= g^{r_3^*},\\
\sigma_4^* &= g^{r_5^*},\\
\sigma_5^* &= g^{r_2^*},
\end{aligned}$$

and $\mathcal{A}$ did not query *Join-User* on inputs ID_g^* and ID^*, did not query *Revoke-User* on inputs ID^* and $RL^{t-1}_{ID_g^*}$, and did not query *Sign* on inputs ID_g^*, ID^* and $\mathfrak{M}^*$, where the identity ID^* belongs to the group named by the identity ID_g^* and $ID^* \notin U^A \setminus RL^t_{ID_g^*}$.

If $\ell \cdot H(ID^*) \neq 0 \ \mathbf{mod}\ q$, or $\nu \cdot H(ID_g^*) \neq 0 \ \mathbf{mod}\ q$ or $\alpha \cdot H(\mathfrak{M}^*) \neq 0 \ \mathbf{mod}\ q$, then the algorithm $\mathcal{B}$ will abort.

If $\ell \cdot H(ID^*) = 0 \ \mathbf{mod}\ q$, and $\nu \cdot H(ID_g^*) = 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}^*) = 0 \ \mathbf{mod}\ q$, then the algorithm $\mathcal{B}$ computes and outputs

$$\begin{aligned}
&\sqrt[2]{\frac{\sigma_0^*}{g_1^{-x_1} \cdot g_3^{-x_2} \cdot g^{r_2^* \cdot H(ID^*)} \cdot g^{r_2^* \cdot \partial} \cdot g^{r_3^* \cdot H(ID_g^*)} \cdot g^{r_3^* \cdot \lambda} \cdot g^{r_4^* \cdot \eta} \cdot g^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot g^{r_5^* \cdot \pi}}}\\
&= \sqrt[2]{\frac{g_2^a \cdot g_4^b \cdot \vartheta^{r_2^* \cdot H(ID^*)} \cdot \psi^{r_2^*} \cdot \mu^{r_3^* \cdot H(ID_g^*)} \cdot \tau^{r_3^*} \cdot \varpi^{r_4^*} \cdot \chi^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot \kappa^{r_5^*}}{g_1^{-x_1} \cdot g_3^{-x_2} \cdot g^{r_2^* \cdot H(ID^*)} \cdot g^{r_2^* \cdot \partial} \cdot g^{r_3^* \cdot H(ID_g^*)} \cdot g^{r_3^* \cdot \lambda} \cdot g^{r_4^* \cdot \eta} \cdot g^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot g^{r_5^* \cdot \pi}}}\\
&= \sqrt[2]{\frac{(g^b \cdot g^{-x_1})^a \cdot (g^a \cdot g^{-x_2})^b \cdot (g_2^\ell \cdot g)^{r_2^* H(ID^*)} \cdot (g^\partial)^{r_2^*} \cdot (g_4^\nu \cdot g)^{r_3^* H(ID_g^*)} \cdot (g^\lambda)^{r_3^*} \cdot (g^\eta)^{r_4^*} \cdot (g_2^\alpha \cdot g)^{r_5^* H(\mathfrak{M}^*)} \cdot (g^\pi)^{r_5^*}}{g_1^{-x_1} \cdot g_3^{-x_2} \cdot g^{r_2^* \cdot H(ID^*)} \cdot g^{r_2^* \cdot \partial} \cdot g^{r_3^* \cdot H(ID_g^*)} \cdot g^{r_3^* \cdot \lambda} \cdot g^{r_4^* \cdot \eta} \cdot g^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot g^{r_5^* \cdot \pi}}}\\
&= g^{a \cdot b},
\end{aligned}$$

which is the solution to the given CDH problem.

Now, we analyze the probability of the algorithm $\mathcal{B}$ not aborting. For the simulation to complete without aborting, we require that all *Join-User* queries will have $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$, all *Revoke-User* queries will have $\ell \cdot H(ID_{u_i^A}) \neq 0 \ \mathbf{mod}\ q$, and all *Sign* queries will have $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}) \neq 0 \ \mathbf{mod}\ q$, and that $\ell \cdot H(ID^*) = 0 \ \mathbf{mod}\ q$, and $\nu \cdot H(ID_g^*) = 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}^*) = 0 \ \mathbf{mod}\ q$ in forgery. If the algorithm $\mathcal{B}$ does not abort, then the following conditions must hold:

(a) $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$ in *Join-User* queries, with $i = 1, 2 \ldots q_j$;
(b) $\ell \cdot H(ID_{u_i^A}) \neq 0 \ \mathbf{mod}\ q$ in *Revoke-User* queries, with $i = 1, 2 \ldots q_r$;
(c) $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}_i) \neq 0 \ \mathbf{mod}\ q$ in *Sign* queries, with $i = 1, 2 \ldots q_s$;
(d) the algorithm $\mathcal{B}$ does not abort in forgery, namely $\ell \cdot H(ID^*) = 0 \ \mathbf{mod}\ q$, and $\nu \cdot H(ID_g^*) = 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}^*) = 0 \ \mathbf{mod}\ q$.

Then we will define the events F_i, E_i, T_i, L_i, R^*, F^*, S^* as

F_i: $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$, with $i = 1, 2 \ldots q_j$;
E_i: $\ell \cdot H(ID_{u_i^A}) \neq 0 \ \mathbf{mod}\ q$, with $i = 1, 2 \ldots q_r$;
T_i: $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$, with $i = 1, 2 \ldots q_s$;

L_i: $\alpha \cdot H(\mathfrak{M}_i) \neq 0 \ \mathbf{mod}\ q$, with $i = 1, 2 \ldots q_s$;
R^*: $\ell \cdot H(ID^*) = 0 \ \mathbf{mod}\ q$;
F^*: $\nu \cdot H(ID_g^*) = 0 \ \mathbf{mod}\ q$;
S^*: $\alpha \cdot H(\mathfrak{M}^*) = 0 \ \mathbf{mod}\ q$.

The probability of $\mathcal{B}$ not aborting is

$$\Pr(not\_abort) = \Pr\left(\bigcap_{i=1}^{q_j} F_i \wedge \bigcap_{i=1}^{q_r} E_i \wedge \bigcap_{i=1}^{q_s} (T_i \wedge L_i) \wedge R^* \wedge F^* \wedge S^*\right).$$

It is easy to see that the events $\bigcap_{i=1}^{q_j} F_i$, $\bigcap_{i=1}^{q_r} E_i$, $\bigcap_{i=1}^{q_s} T_i$, $\bigcap_{i=1}^{q_s} L_i$, R^*, F^* and S^* are independent. Then we may compute

$$\Pr\left(\bigcap_{i=1}^{q_j} F_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_j} \neg F_i\right) = 1 - q_j \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_j}{q};$$
$$\Pr\left(\bigcap_{i=1}^{q_r} E_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_r} \neg E_i\right) = 1 - q_r \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_r}{q};$$
$$\Pr\left(\bigcap_{i=1}^{q_s} T_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_s} \neg T_i\right) = 1 - q_s \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_s}{q};$$
$$\Pr\left(\bigcap_{i=1}^{q_s} L_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_s} \neg L_i\right) = 1 - q_s \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_s}{q};$$
$$\Pr(R^*) = \frac{1^k}{1^k \cdot q} = \frac{1}{q}; \quad \Pr(F^*) = \frac{1^k}{1^k \cdot q} = \frac{1}{q}; \quad \Pr(S^*) = \frac{1^k}{1^k \cdot q} = \frac{1}{q}.$$

Thus,

$$\begin{aligned}
\Pr(not_abort) &= \Pr\left(\bigcap_{i=1}^{q_j} F_i \wedge \bigcap_{i=1}^{q_r} E_i \wedge \bigcap_{i=1}^{q_s} (T_i \wedge L_i) \wedge R^* \wedge F^* \wedge S^*\right) \\
&= \Pr\left(\bigcap_{i=1}^{q_j} F_i\right) \cdot \Pr\left(\bigcap_{i=1}^{q_r} E_i\right) \cdot \Pr\left(\bigcap_{i=1}^{q_s} T_i\right) \cdot \Pr\left(\bigcap_{i=1}^{q_s} L_i\right) \cdot \Pr(R^*) \cdot \Pr(F^*) \cdot \Pr(S^*) \\
&= \left(1 - \frac{q_j}{q}\right) \cdot \left(1 - \frac{q_r}{q}\right) \cdot \left(1 - \frac{q_s}{q}\right)^2 \cdot \frac{1}{q^3}.
\end{aligned}$$

So we can get that $\varepsilon' = (1 - \frac{q_j}{q}) \cdot (1 - \frac{q_r}{q}) \cdot (1 - \frac{q_s}{q})^2 \cdot \frac{\varepsilon}{q^3}$.

If the simulation does not abort, the adversary $\mathcal{A}$ will create a valid signature forgery with probability at least ε. The algorithm $\mathcal{B}$ can then compute $g^{a \cdot b}$ from the forgery as shown above. The time complexity of the algorithm $\mathcal{B}$ is dominated by the time for the exponentiations and multiplications in the queries. We assume that the time for integer addition and integer multiplication, and the time for hash computation can both be ignored, then the time complexity of the algorithm $\mathcal{B}$ is

$$\begin{aligned}
\hbar' = \hbar + O(&q_j \cdot (10 \cdot C_{exp} + 7 \cdot C_{mul_1} + 1 \cdot C_{pair}) + q_r \cdot (6 \cdot C_{exp} + 3 \cdot C_{mul_1} + 2 \cdot C_{pair} + C_{mul_2}) \\
&+ q_s \cdot (15 \cdot C_{exp} + 12 \cdot C_{mul_1} + 1 \cdot C_{pair})).
\end{aligned}$$

c) *Framing attacks*:

Let **TIBGS** be a traceable identity-based group signature scheme of Section 6. Additionally, let $\mathcal{A}$ be an ($\hbar$, ε, q_g, q_j, q_r, q_s)-adversary attacking **TIBGS**. From the adversary $\mathcal{A}$, we construct an algorithm $\mathcal{B}$, for (g, g^a, g^b)$\in \mathbb{G}_1$, the algorithm $\mathcal{B}$ is able to use $\mathcal{A}$ to compute $g^{a\cdot b}$. Thus, we assume the algorithm $\mathcal{B}$ can solve the CDH with probability at least ε' and in time at most $\hbar'$, contradicting the ($\hbar'$, ε')-CDH assumption. According to the algorithm $\mathcal{B}_{TF\_TIBGS}$ of Definition 5.2, such a simulation may be created in the following way (to avoid the symbol confused, we use u_i^B and U^B to replace u_i^b and U^b of the algorithm $\mathcal{B}_{TF\_TIBGS}$):

Setup: The PKG system inputs a security parameter 1^k. Additionally, let $\mathbb{G}_1$ and $\mathbb{G}_2$ be groups of prime order q and g be a generator of $\mathbb{G}_1$, and let $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ denote the bilinear map. The size of the group is determined by the security parameter, and we set $\mathbb{A} \subseteq \mathbb{Z}_q$ as the universe of identities. One hash function, $H : \{0,1\}^* \to \mathbb{Z}_{1^k \cdot q}$ can be defined and used to generate any integer value in $\mathbb{Z}_{1^k \cdot q}$ (where 1^k represents the corresponding decimal number).

Then the system parameters are generated as follows. The algorithm chooses random $x_1, x_2 \in \mathbb{Z}_q$, and then sets $g_1 = g^a$, $g_2 = g^b \cdot g^{-x_1}$, $g_3 = g^b$ and $g_4 = g^a \cdot g^{-x_2}$ ($\mathcal{B}$ doesn't know a and b). Also the algorithm chooses ℓ, ∂, ν, λ, η, α and $\pi \in \mathbb{Z}_q$, and then sets $\vartheta = g_2^{\ell} \cdot g$, $\psi = g^{\partial}$, $\mu = g_4^{\nu} \cdot g$, $\tau = g^{\lambda}$, $\varpi = g^{\eta}$, $\chi = g_2^{\alpha} \cdot g$ and $\kappa = g^{\pi}$. Finally, the system outputs the public parameters $TIBGK$=($\mathbb{G}_1$, $\mathbb{G}_2$, e, g, g_1, g_2, g_3, g_4, ϑ, ψ, μ, τ, ϖ, χ, κ).

Additionally, because the algorithm $\mathcal{B}$ doesn't know a and b, the algorithm can construct all private keys of users by the following computation: for one user u (ID is the identity of the user u), the algorithm $\mathcal{B}$ chooses a random $r_1 \in \mathbb{Z}_q$ and computes $x_0 = g_1^{-\frac{1}{\ell}} \cdot \vartheta^{r_1} \cdot g_1^{-\frac{\partial}{\ell}\cdot\frac{1}{H(ID)}} \cdot \psi^{\frac{r_1}{H(ID)}}$, $x_1 = (g_1^{-\frac{1}{\ell}} \cdot g^{r_1})^{\frac{1}{H(ID)}}$, and then outputs a private key $sk_{\{ID\}} = \{x_0, x_1\}$ to $\mathcal{A}$. Similarly, setting $r_1' = (r_1 - \frac{a}{\ell}) \cdot \frac{1}{H(ID)}$, $sk_{\{ID\}} = \{x_0, x_1\} = \{g_2^a \cdot \vartheta^{r_1' \cdot H(ID)} \cdot \psi^{r_1'},\ \ g^{r_1'}\}$ is a valid private key, where we assure that $\ell \cdot H(ID) \neq 0$ **mod** q.

Queries: When running the adversary $\mathcal{A}$, the relevant queries can occur according to the algorithm $\mathcal{B}_{TF\_TIBGS}$ of Definition 5.2. The algorithm $\mathcal{B}$ answers these in the following way:

- *Group-Setup queries*: Given the public parameters $TIBGK$ and the identity ID_g of the group, the algorithm $\mathcal{B}$ similarly constructs a group private key $gsk_{\{ID_g\}} = \{y_0,\ y_1\} = \{g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu}\cdot\frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}},\ \ (g_3^{-\frac{1}{\nu}} \cdot g^{r_2})^{\frac{1}{H(ID_g)}}\}$ to the adversary $\mathcal{A}$. Setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$, $gsk_{\{ID_g\}} = \{y_0,\ y_1\} = \{g_4^b \cdot \mu^{r_2' \cdot H(ID_g)} \cdot \tau^{r_2'},\ \ g^{r_2'}\}$ is a valid private key, where we assure that $\nu \cdot H(ID_g) \neq 0$ **mod** q.
- *Join-User queries*: Given the public parameters $TIBGK$, the identity ID_g of the group and the identity $ID_{u_i^B}$ of the group member (the user u_i^B is added to the set U^B where $U^B \neq \varnothing$), the algorithm chooses random $r_2, r_3, r_4 \in \mathbb{Z}_q$ and computes

$$\begin{aligned}
v_0 &= g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu}\cdot\frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}} \cdot \vartheta^{r_3 \cdot H(ID_{u_i^B})} \cdot \psi^{r_3} \cdot \varpi^{r_4}, \\
v_1 &= e(\vartheta^{r_3 \cdot H(ID_{u_i^B})} \cdot \psi^{r_3}, g), \\
v_2 &= g^{r_4},\ v_3 = (g_3^{-\frac{1}{\nu}} \cdot g^{r_2})^{\frac{1}{H(ID_g)}},\ v_4 = g^{r_3}.
\end{aligned}$$

Finally, the algorithm outputs a member private key $usk_{\{ID_{u_i^B}\}} = \{v_0, v_1, v_2, v_3, v_4\}$ to the adversary $\mathcal{A}$. Similarly, setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$,

$$\begin{aligned}
usk_{\{ID_{u_i^B}\}} &= \{v_0, v_1, v_2, v_3, v_4\} \\
&= \{g_4^b \cdot \mu^{r_2' \cdot H(ID_g)} \cdot \tau^{r_2'} \cdot \vartheta^{r_3 \cdot H(ID_{u_i^B})} \cdot \psi^{r_3} \cdot \varpi^{r_4},\ \ e(\vartheta^{r_3 \cdot H(ID_{u_i^B})} \cdot \psi^{r_3}, g),\ \ g^{r_4},\ \ g^{r_2'},\ \ g^{r_3}\}.
\end{aligned}$$

So, $usk_{\{ID_{u_i^B}\}}$ is a valid private key, where we assure that $\nu \cdot H(ID_g) \neq 0$ **mod** q.

Remark B.5. Where we maximize the adversary's advantage, thus v_4 is also passed to $\mathcal{A}$.

- *Revoke-User*: Given the public parameters $TIBGK$, the identity $ID_{u_i^B}$ of the revoked group member and the revocation list RL_{ID}^t of the last duration t ($RL_{ID}^t = \varnothing$ when $t = 0$), the algorithm chooses random $r_1, r_3, r_4 \in \mathbb{Z}_q$ and computes

$$T = e\left(\vartheta^{H(ID_{u_i^B})} \cdot \psi,\ \left(g_1^{-\frac{1}{\ell}} \cdot g^{r_1}\right)^{\frac{1}{H(ID_{u_i^B})}}\right) \cdot e\left(\vartheta^{r_3 \cdot H(ID_{u_i^B})} \cdot \psi^{r_3}, g\right),$$

$$\upsilon_2 = g^{r_4}.$$

Finally, the algorithm outputs and adds a tuple $[ID_{u_i^B}, T, \upsilon_2]$ to the revocation list RL_{ID}^t, and then an updated revocation list RL_{ID}^{t+1} is published to the adversary $\mathcal{A}$. Similarly, setting $r_1' = (r_1 - \frac{a}{\ell}) \cdot \frac{1}{H(ID_{u_i^B})}$,

$$\begin{aligned} T &= e\left(\vartheta^{H(ID_{u_i^B})} \cdot \psi,\ \left(g_1^{-\frac{1}{\ell}} \cdot g^{r_1}\right)^{\frac{1}{H(ID_{u_i^B})}}\right) \cdot e\left(\vartheta^{r_3 \cdot H(ID_{u_i^B})} \cdot \psi^{r_3}, g\right) \\ &= e\left(\vartheta^{H(ID_{u_i^B})} \cdot \psi,\ g^{(r_1 - \frac{a}{\ell}) \cdot \frac{1}{H(ID_{u_i^B})}}\right) \cdot e\left(\vartheta^{r_3 \cdot H(ID_{u_i^B})} \cdot \psi^{r_3}, g\right) \\ &= e\left(\vartheta^{H(ID_{u_i^B})} \cdot \psi, g^{r_1'}\right) \cdot e\left(\vartheta^{r_3 \cdot H(ID_{u_i^B})} \cdot \psi^{r_3}, g\right) \\ &= e\left(\vartheta^{(r_1' + r_3) \cdot H(ID_{u_i^B})} \cdot \psi^{(r_1' + r_3)}, g\right), \end{aligned}$$

thus the tuple $[ID_{u_i^B}, T, \upsilon_2]$ is a valid data, where we assure that $\ell \cdot H(ID_{u_i^B}) \neq 0 \ \mathbf{mod}\ q$.

- *Sign Queries*: given the public parameters $TIBGK$, the identity ID_g of the group, the identity $ID_{u_i^B}$ of the group member (the user u_i^B is added to the set U^B) and the message $\mathfrak{M}$, the algorithm chooses random $r_2, r_3, r_4, r_5 \in \mathbb{Z}_q$ and computes

$$\sigma_0 = g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu} \cdot \frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}} \cdot \vartheta^{r_3 \cdot H(ID_{u_i^B})} \cdot \psi^{r_3} \cdot \varpi^{r_4} \cdot g_1^{-\frac{1}{\alpha}} \cdot \chi^{r_5} \cdot g_1^{-\frac{\pi}{\alpha} \cdot \frac{1}{H(\mathfrak{M})}} \cdot \kappa^{\frac{r_5}{H(\mathfrak{M})}},$$

$$\sigma_1 = e(\vartheta^{r_3 \cdot H(ID_{u_i^B})} \cdot \psi^{r_3}, g),$$

$$\sigma_2 = g^{r_4},$$

$$\sigma_3 = (g_3^{-\frac{1}{\nu}} \cdot g^{r_2})^{\frac{1}{H(ID_g)}},$$

$$\sigma_4 = (g_1^{-\frac{1}{\alpha}} \cdot g^{r_5})^{\frac{1}{H(\mathfrak{M})}},$$

$$\sigma_5 = g^{r_3}.$$

Finally, the algorithm outputs a signature $\Phi = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$ to the adversary $\mathcal{A}$. Similarly, we maximize the adversary's advantage, thus σ_5 is also passed to the adversary $\mathcal{A}$. Setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$ and $r_5' = (r_5 - \frac{a}{\alpha}) \cdot \frac{1}{H(\mathfrak{M})}$, $\Phi = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$ is a valid signature, where we assure that $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}) \neq 0 \ \mathbf{mod}\ q$.

Forgery: If the algorithm $\mathcal{B}$ does not abort as a consequence of one of the queries above, the adversary $\mathcal{A}$ will, with probability at least ε, return a message $\mathfrak{M}^*$, and a valid identity-based group signature forgery, $\Phi^* = \{\sigma_0^*, \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*\}$ for the identity ID^* of the group member, the identity ID_g^* of the group and the revocation list $RL^t_{ID^*_g}$, where

$$\begin{aligned}
\sigma_0^* &= g_2^a \cdot g_4^b \cdot \vartheta^{r_2^* \cdot H(ID^*)} \cdot \psi^{r_2^*} \cdot \mu^{r_3^* \cdot H(ID_g^*)} \cdot \tau^{r_3^*} \cdot \varpi^{r_4^*} \cdot \chi^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot \kappa^{r_5^*},\\
\sigma_1^* &= e(\vartheta^{r_2^* \cdot H(ID^*)} \cdot \psi^{r_2^*}, g),\\
\sigma_2^* &= g^{r_4^*},\\
\sigma_3^* &= g^{r_3^*},\\
\sigma_4^* &= g^{r_5^*},\\
\sigma_5^* &= g^{r_2^*}.
\end{aligned}$$

And $\mathcal{A}$ did not query *Group-Setup* on input ID_g^*, did not query *Join-User* on inputs ID_g^* and ID^*, did not query *Revoke-User* on inputs ID^* and $RL^{t-1}_{ID^*_g}$, and did not query *Sign* on inputs ID_g^*, ID^* and $\mathfrak{M}^*$, where the identity ID^* belongs to the group named by the identity ID_g^* and $ID^* \in U^B$.

If $\ell \cdot H(ID^*) \neq 0 \ \mathbf{mod}\ q$, or $\nu \cdot H(ID_g^*) \neq 0 \ \mathbf{mod}\ q$ or $\alpha \cdot H(\mathfrak{M}^*) \neq 0 \ \mathbf{mod}\ q$, then the algorithm $\mathcal{B}$ will abort.

If $\ell \cdot H(ID^*) = 0 \ \mathbf{mod}\ q$, and $\nu \cdot H(ID_g^*) = 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}^*) = 0 \ \mathbf{mod}\ q$, then the algorithm $\mathcal{B}$ computes and outputs

$$\begin{aligned}
&\sqrt[2]{\frac{\sigma_0^*}{g_1^{-x_1} \cdot g_3^{-x_2} \cdot g^{r_2^* \cdot H(ID^*)} \cdot g^{r_2^* \cdot \partial} \cdot g^{r_3^* \cdot H(ID_g^*)} \cdot g^{r_3^* \cdot \lambda} \cdot g^{r_4^* \cdot \eta} \cdot g^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot g^{r_5^* \cdot \pi}}}\\
&= \sqrt[2]{\frac{g_2^a \cdot g_4^b \cdot \vartheta^{r_2^* \cdot H(ID^*)} \cdot \psi^{r_2^*} \cdot \mu^{r_3^* \cdot H(ID_g^*)} \cdot \tau^{r_3^*} \cdot \varpi^{r_4^*} \cdot \chi^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot \kappa^{r_5^*}}{g_1^{-x_1} \cdot g_3^{-x_2} \cdot g^{r_2^* \cdot H(ID^*)} \cdot g^{r_2^* \cdot \partial} \cdot g^{r_3^* \cdot H(ID_g^*)} \cdot g^{r_3^* \cdot \lambda} \cdot g^{r_4^* \cdot \eta} \cdot g^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot g^{r_5^* \cdot \pi}}}\\
&= \sqrt[2]{\frac{(g^b \cdot g^{-x_1})^a \cdot (g^a \cdot g^{-x_2})^b \cdot (g_2^\ell \cdot g)^{r_2^* H(ID^*)} \cdot (g^\partial)^{r_2^*} \cdot (g_4^\nu \cdot g)^{r_3^* H(ID_g^*)} \cdot (g^\lambda)^{r_3^*} \cdot (g^\eta)^{r_4^*} \cdot (g_2^\alpha \cdot g)^{r_5^* H(\mathfrak{M}^*)} \cdot (g^\pi)^{r_5^*}}{g_1^{-x_1} \cdot g_3^{-x_2} \cdot g^{r_2^* H(ID^*)} \cdot g^{r_2^* \cdot \partial} \cdot g^{r_3^* \cdot H(ID_g^*)} \cdot g^{r_3^* \cdot \lambda} \cdot g^{r_4^* \cdot \eta} \cdot g^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot g^{r_5^* \cdot \pi}}}\\
&= g^{a \cdot b},
\end{aligned}$$

which is the solution to the given CDH problem.

Now, we analyze the probability of the algorithm $\mathcal{B}$ not aborting. For the simulation to complete without aborting, we require that all *Group-Setup* queries will have $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$, all *Join-User* queries will have $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$, all *Revoke-User* queries will have $\ell \cdot H(ID_{u_i^B}) \neq 0 \ \mathbf{mod}\ q$, and all *Sign* queries will have $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}) \neq 0 \ \mathbf{mod}\ q$, and that $\ell \cdot H(ID^*) = 0 \ \mathbf{mod}\ q$, and $\nu \cdot H(ID_g^*) = 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}^*) = 0 \ \mathbf{mod}\ q$ in forgery. If the algorithm $\mathcal{B}$ does not abort, then the following conditions must hold:

(a) $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$ in *Group-Setup* queries, with $i = 1, 2 \ldots q_g$;
(b) $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$ in *Join-User* queries, with $i = 1, 2 \ldots q_j$;
(c) $\ell \cdot H(ID_{u_i^B}) \neq 0 \ \mathbf{mod}\ q$ in *Revoke-User* queries, with $i = 1, 2 \ldots q_r$;
(d) $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}_i) \neq 0 \ \mathbf{mod}\ q$ in *Sign* queries, with $i = 1, 2 \ldots q_s$;
(e) the algorithm $\mathcal{B}$ does not abort in forgery, namely $\ell \cdot H(ID^*) = 0 \ \mathbf{mod}\ q$, and $\nu \cdot H(ID_g^*) = 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}^*) = 0 \ \mathbf{mod}\ q$.

Then we will define the events D_i, F_i, E_i, T_i, L_i, R^*, F^*, S^* as

D_i: $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$, with $i = 1, 2 \ldots q_g$;
F_i: $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$, with $i = 1, 2 \ldots q_j$;
E_i: $\ell \cdot H(ID_{u_i^B}) \neq 0 \ \mathbf{mod}\ q$, with $i = 1, 2 \ldots q_r$;
T_i: $\nu \cdot H(ID_{g_i}) \neq 0 \ \mathbf{mod}\ q$, with $i = 1, 2 \ldots q_s$;

L_i: $\alpha \cdot H(\mathfrak{M}_i) \neq 0$ **mod** q, with $i = 1, 2 \ldots q_s$;
R^*: $\ell \cdot H(ID^*) = 0$ **mod** q;
F^*: $\nu \cdot H(ID_g^*) = 0$ **mod** q;
S^*: $\alpha \cdot H(\mathfrak{M}^*) = 0$ **mod** q.

The probability of $\mathcal{B}$ not aborting is

$$\Pr(not\_abort) = \Pr\left(\bigcap_{i=1}^{q_g} D_i \wedge \bigcap_{i=1}^{q_j} F_i \wedge \bigcap_{i=1}^{q_r} E_i \wedge \bigcap_{i=1}^{q_s} (T_i \wedge L_i) \wedge R^* \wedge F^* \wedge S^*\right).$$

It is easy to see that the events $\bigcap_{i=1}^{q_g} D_i$, $\bigcap_{i=1}^{q_j} F_i$, $\bigcap_{i=1}^{q_r} E_i$, $\bigcap_{i=1}^{q_s} T_i$, $\bigcap_{i=1}^{q_s} L_i$, R^*, F^* and S^* are independent. Then we may compute

$$\Pr\left(\bigcap_{i=1}^{q_g} D_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_g} \neg D_i\right) = 1 - q_g \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_g}{q};$$

$$\Pr\left(\bigcap_{i=1}^{q_j} F_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_j} \neg F_i\right) = 1 - q_j \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_j}{q};$$

$$\Pr\left(\bigcap_{i=1}^{q_r} E_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_r} \neg E_i\right) = 1 - q_r \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_r}{q};$$

$$\Pr\left(\bigcap_{i=1}^{q_s} T_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_s} \neg T_i\right) = 1 - q_s \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_s}{q};$$

$$\Pr\left(\bigcap_{i=1}^{q_s} L_i\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_s} \neg L_i\right) = 1 - q_s \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_s}{q};$$

$$\Pr(R^*) = \frac{1^k}{1^k \cdot q} = \frac{1}{q}; \quad \Pr(F^*) = \frac{1^k}{1^k \cdot q} = \frac{1}{q}; \quad \Pr(S^*) = \frac{1^k}{1^k \cdot q} = \frac{1}{q}\cdot$$

Thus,

$$\begin{aligned}
\Pr(not_abort) &= \Pr\left(\bigcap_{i=1}^{q_g} D_i \wedge \bigcap_{i=1}^{q_j} F_i \wedge \bigcap_{i=1}^{q_r} E_i \wedge \bigcap_{i=1}^{q_s} (T_i \wedge L_i) \wedge R^* \wedge F^* \wedge S^*\right) \\
&= \Pr\left(\bigcap_{i=1}^{q_g} D_i\right) \cdot \Pr\left(\bigcap_{i=1}^{q_j} F_i\right) \cdot \Pr\left(\bigcap_{i=1}^{q_r} E_i\right) \cdot \Pr\left(\bigcap_{i=1}^{q_s} T_i\right) \cdot \Pr\left(\bigcap_{i=1}^{q_s} L_i\right) \cdot \Pr(R^*) \cdot \Pr(F^*) \cdot \Pr(S^*) \\
&= \left(1 - \frac{q_g}{q}\right) \cdot \left(1 - \frac{q_j}{q}\right) \cdot \left(1 - \frac{q_r}{q}\right) \cdot \left(1 - \frac{q_s}{q}\right)^2 \cdot \frac{1}{q^3}.
\end{aligned}$$

So we can get that $\varepsilon' = (1 - \frac{q_g}{q}) \cdot (1 - \frac{q_j}{q}) \cdot (1 - \frac{q_r}{q}) \cdot (1 - \frac{q_s}{q})^2 \cdot \frac{\varepsilon}{q^3}$.

If the simulation does not abort, the adversary $\mathcal{A}$ will create a valid signature forgery with probability at least ε. The algorithm $\mathcal{B}$ can then compute $g^{a \cdot b}$ from the forgery as shown above. The time complexity of the algorithm $\mathcal{B}$ is dominated by the time for the exponentiations and multiplications in the queries. We similarly assume that the time for integer addition and integer multiplication, and the time for hash computation can both be ignored, then the time complexity of the algorithm $\mathcal{B}$ is

$$\begin{aligned}\hbar' = \hbar + O(&q_g \cdot (5 \cdot C_{exp} + 4 \cdot C_{mul_1}) + q_j \cdot (10 \cdot C_{exp} + 7 \cdot C_{mul_1} + 1 \cdot C_{pair}) \\ &+ q_r \cdot (6 \cdot C_{exp} + 3 \cdot C_{mul_1} + 2 \cdot C_{pair} + C_{mul_2}) + q_s \cdot (15 \cdot C_{exp} + 12 \cdot C_{mul_1} + 1 \cdot C_{pair})).\end{aligned}$$

Therefore, from the above proofs, we may get that

$$\varepsilon'' = \left[\frac{\varepsilon' \cdot q^3}{\left(1 - \frac{q_j}{q}\right) \cdot \left(1 - \frac{q_r}{q}\right) \cdot \left(1 - \frac{q_s}{q}\right)^2}\right] \Bigg\| \left[\frac{\varepsilon' \cdot q^3}{\left(1 - \frac{q_g}{q}\right) \cdot \left(1 - \frac{q_j}{q}\right) \cdot \left(1 - \frac{q_r}{q}\right) \cdot \left(1 - \frac{q_s}{q}\right)^2}\right],$$

$$\begin{aligned}\hbar'' = \mathrm{MAX}\{&\hbar' - O(q_j \cdot (10 \cdot C_{exp} + 7 \cdot C_{mul_1} + 1 \cdot C_{pair}) + q_r \cdot (6 \cdot C_{exp} + 3 \cdot C_{mul_1} + 2 \cdot C_{pair} + C_{mul_2}) \\ &+ q_s \cdot (15 \cdot C_{exp} + 12 \cdot C_{mul_1} + 1 \cdot C_{pair})), \quad \hbar' - O(q_g \cdot (5 \cdot C_{exp} + 4 \cdot C_{mul_1}) \\ &+ q_j \cdot (10 \cdot C_{exp} + 7 \cdot C_{mul_1} + 1 \cdot C_{pair}) + q_r \cdot (6 \cdot C_{exp} + 3 \cdot C_{mul_1} + 2 \cdot C_{pair} + C_{mul_2}) \\ &+ q_s \cdot (15 \cdot C_{exp} + 12 \cdot C_{mul_1} + 1 \cdot C_{pair}))\}.\end{aligned}$$

Thus, Theorem 7.2 follows. □

Proof of Theorem 7.3

Proof. Let **TIBGS** be a traceable identity-based group signature scheme of Section 6. Additionally, let $\mathcal{A}$ be an ($\hbar$, ε, q_g, q_j, q_r, q_s)-adversary attacking **TIBGS**. From the adversary $\mathcal{A}$, we construct an algorithm $\mathcal{B}$, for (g, g^a, g^b)$\in \mathbb{G}_1$, the algorithm $\mathcal{B}$ is able to use $\mathcal{A}$ to compute $g^{a \cdot b}$. Thus, we assume the algorithm $\mathcal{B}$ can solve the CDH with probability at least ε' and in time at most $\hbar'$, contradicting the ($\hbar'$, ε')-CDH assumption. According to the algorithm $\mathcal{B}_{A\_TIBGS}$ of Definition 5.3, such a simulation may be created in the following way:

Setup: The PKG system inputs a security parameter 1^k. Additionally, let $\mathbb{G}_1$ and $\mathbb{G}_2$ be groups of prime order q and g be a generator of $\mathbb{G}_1$, and let $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ denote the bilinear map. The size of the group is determined by the security parameter, and we set $\mathbb{A} \subseteq \mathbb{Z}_q$ as the universe of identities. One hash function, $H : \{0,1\}^* \to \mathbb{Z}_{1^k \cdot q}$ can be defined and used to generate any integer value in $\mathbb{Z}_{1^k \cdot q}$ (where 1^k represents the corresponding decimal number).

Then the system parameters are generated as follows. The algorithm chooses random $x_1, x_2 \in \mathbb{Z}_q$, and then sets $g_1 = g^a$, $g_2 = g^b \cdot g^{-x_1}$, $g_3 = g^b$ and $g_4 = g^a \cdot g^{-x_2}$ ($\mathcal{B}$ doesn't know a and b). Also the algorithm chooses ℓ, ∂, ν, λ, η, α and $\pi \in \mathbb{Z}_q$, and then sets $\vartheta = g_2^{\ell} \cdot g$, $\psi = g^{\partial}$, $\mu = g_4^{\nu} \cdot g$, $\tau = g^{\lambda}$, $\varpi = g^{\eta}$, $\chi = g_2^{\alpha} \cdot g$ and $\kappa = g^{\pi}$. Finally, the system outputs the public parameters $TIBGK$=($\mathbb{G}_1$, $\mathbb{G}_2$, e, g, g_1, g_2, g_3, g_4, ϑ, ψ, μ, τ, ϖ, χ, κ).

Additionally, because the algorithm $\mathcal{B}$ doesn't know a and b, the algorithm can construct all private keys of users by the following computation: for one user u (ID is the identity of the user u), the algorithm $\mathcal{B}$ chooses a random $r_1 \in \mathbb{Z}_q$ and computes $x_0 = g_1^{-\frac{1}{\ell}} \cdot \vartheta^{r_1} \cdot g_1^{-\frac{\partial}{\ell} \cdot \frac{1}{H(ID)}} \cdot \psi^{\frac{r_1}{H(ID)}}$, $x_1 = (g_1^{-\frac{1}{\ell}} \cdot g^{r_1})^{\frac{1}{H(ID)}}$, and then outputs a private key $sk_{\{ID\}} = \{x_0, x_1\}$ to $\mathcal{A}$. Similarly, setting $r_1' = (r_1 - \frac{a}{\ell}) \cdot \frac{1}{H(ID)}$, $sk_{\{ID\}} = \{x_0, x_1\} = \{g_2^a \cdot \vartheta^{r_1' \cdot H(ID)} \cdot \psi^{r_1'},\ g^{r_1'}\}$ is a valid private key, where we assure that $\ell \cdot H(ID) \neq 0$ **mod** q.

Queries Phase 1: When running the adversary $\mathcal{A}$, the relevant queries can occur according to the algorithm $\mathcal{B}_{A\_TIBGS}$ of Definition 5.3. The algorithm $\mathcal{B}$ answers these in the following way:

- *Group-Setup queries*: Given the public parameters $TIBGK$ and the identity ID_g of the group, the algorithm $\mathcal{B}$ similarly constructs a group private key $gsk_{\{ID_g\}} = \{y_0,\ y_1\} = \{g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu} \cdot \frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}},\ (g_3^{-\frac{1}{\nu}} \cdot g^{r_2})^{\frac{1}{H(ID_g)}}\}$ to the adversary $\mathcal{A}$. Setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$, $gsk_{\{ID_g\}} = \{y_0,\ y_1\} = \{g_4^b \cdot \mu^{r_2' \cdot H(ID_g)} \cdot \tau^{r_2'},\ g^{r_2'}\}$ is a valid private key, where we assure that $\nu \cdot H(ID_g) \neq 0$ **mod** q.

- *Join-User queries*: Given the public parameters $TIBGK$, the identity ID_g of the group and the identity ID_i of the group member, the algorithm chooses random $r_2, r_3, r_4 \in \mathbb{Z}_q$ and computes

$$v_0 = g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu} \cdot \frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}} \cdot \vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3} \cdot \varpi^{r_4},$$
$$v_1 = e\left(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g\right),$$
$$v_2 = g^{r_4},\ v_3 = (g_3^{-\frac{1}{\nu}} \cdot g^{r_2})^{\frac{1}{H(ID_g)}},\ v_4 = g^{r_3}.$$

Finally, the algorithm outputs a member private key $usk_{\{ID_i\}} = \{v_0, v_1, v_2, v_3, v_4\}$ to the adversary $\mathcal{A}$. Similarly, setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$, $usk_{\{ID_i\}} = \{v_0, v_1, v_2, v_3, v_4\} = \{g_4^b \cdot \mu^{r_2' \cdot H(ID_g)} \cdot \tau^{r_2'} \cdot \vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3} \cdot \varpi^{r_4},\ e(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g),\ g^{r_4},\ g^{r_2'},\ g^{r_3}\}$ is a valid private key, where we assure that $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$.

Remark B.6. Where we maximize the adversary's advantage, thus v_4 is also passed to $\mathcal{A}$.

- *Revoke-User*: Given the public parameters $TIBGK$, the identity ID_i of the revoked group member and the revocation list RL_{ID}^t of the last duration t ($RL_{ID}^t = \varnothing$ when $t = 0$), the algorithm chooses random $r_1, r_3, r_4 \in \mathbb{Z}_q$ and computes

$$T = e(\vartheta^{H(ID_i)} \cdot \psi,\ (g_1^{-\frac{1}{\ell}} \cdot g^{r_1})^{\frac{1}{H(ID_i)}}) \cdot e(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g),$$
$$v_2 = g^{r_4}.$$

Finally, the algorithm outputs and adds a tuple $[ID_i, T, v_2]$ to the revocation list RL_{ID}^t, and then an updated revocation list RL_{ID}^{t+1} is published to the adversary $\mathcal{A}$. Similarly, setting $r_1' = (r_1 - \frac{a}{\ell}) \cdot \frac{1}{H(ID_i)}$,

$$\begin{aligned} T &= e\left(\vartheta^{H(ID_i)} \cdot \psi,\ \left(g_1^{-\frac{1}{\ell}} \cdot g^{r_1}\right)^{\frac{1}{H(ID_i)}}\right) \cdot e\left(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g\right) \\ &= e\left(\vartheta^{H(ID_i)} \cdot \psi,\ g^{(r_1 - \frac{a}{\ell}) \cdot \frac{1}{H(ID_i)}}\right) \cdot e\left(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g\right) \\ &= e\left(\vartheta^{H(ID_i)} \cdot \psi, g^{r_1'}\right) \cdot e\left(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g\right) \\ &= e\left(\vartheta^{(r_1' + r_3) \cdot H(ID_i)} \cdot \psi^{(r_1' + r_3)}, g\right), \end{aligned}$$

thus the tuple $[ID_i, T, v_2]$ is a valid data, where we assure that $\ell \cdot H(ID_i) \neq 0 \ \mathbf{mod}\ q$.

- *Sign queries*: Given the public parameters $TIBGK$, the identity ID_g of the group, the identity ID_i of the group member and the message $\mathfrak{M}$, the algorithm chooses random $r_2, r_3, r_4, r_5 \in \mathbb{Z}_q$ and computes

$$\sigma_0 = g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu} \cdot \frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}} \cdot \vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3} \cdot \varpi^{r_4} \cdot g_1^{-\frac{1}{\alpha}} \cdot \chi^{r_5} \cdot g_1^{-\frac{\pi}{\alpha} \cdot \frac{1}{H(\mathfrak{M})}} \cdot \kappa^{\frac{r_5}{H(\mathfrak{M})}},$$
$$\sigma_1 = e(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g),$$
$$\sigma_2 = g^{r_4},$$
$$\sigma_3 = (g_3^{-\frac{1}{\nu}} \cdot g^{r_2})^{\frac{1}{H(ID_g)}},$$
$$\sigma_4 = (g_1^{-\frac{1}{\alpha}} \cdot g^{r_5})^{\frac{1}{H(\mathfrak{M})}},$$
$$\sigma_5 = g^{r_3}.$$

Finally, the algorithm outputs a signature $\Phi = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$ to the adversary $\mathcal{A}$. Similarly, we maximize the adversary's advantage, thus σ_5 is also passed to the adversary $\mathcal{A}$. Setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$ and $r_5' = (r_5 - \frac{a}{\alpha}) \cdot \frac{1}{H(\mathfrak{M})}$, $\Phi = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$ is a valid signature, where we assure that $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}) \neq 0 \ \mathbf{mod}\ q$.

Challenge: If the algorithm $\mathcal{B}$ does not abort as a consequence of one of the queries above, the adversary $\mathcal{A}$ will send its forgery ($\mathfrak{M}^*$, ID_g^*, $RL^t_{ID_g^*}$) and two group member identities ID_0^* and ID_1^* that belong to the group named by the group identity ID_g^* to the challenger. The forgery satisfies the following conditions:

(a) $\mathcal{A}$ did not query *Group-Setup* on input ID_g^*;
(b) $\mathcal{A}$ did not query *Join-User* on inputs ID_g^*, ID_0^* (and ID_1^*);
(c) $\mathcal{A}$ did not query *Revoke-User* on inputs ID_g^*, ID_0^* (and ID_1^*) and $RL^{t-1}_{ID_g^*}$.

The challenger picks a random bit $x \in \{0,1\}$, and runs $\Phi^* = \{\sigma_0^*, \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*\}$ $\leftarrow$*Sign*($TIBGK$, $sk_{ID_x^*}$, $\mathfrak{M}^*$), and then outputs $\Phi^* = \{\sigma_0^*, \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*\}$ to $\mathcal{A}$, where

$$\begin{aligned}
\sigma_0^* &= g_2^a \cdot g_4^b \cdot \vartheta^{r_2^* \cdot H(ID_x^*)} \cdot \psi^{r_2^*} \cdot \mu^{r_3^* \cdot H(ID_g^*)} \cdot \tau^{r_3^*} \cdot \varpi^{r_4^*} \cdot \chi^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot \kappa^{r_5^*},\\
\sigma_1^* &= e(\vartheta^{r_2^* \cdot H(ID_x^*)} \cdot \psi^{r_2^*}, g),\\
\sigma_2^* &= g^{r_4^*},\\
\sigma_3^* &= g^{r_3^*},\\
\sigma_4^* &= g^{r_5^*},\\
\sigma_5^* &= g^{r_2^*}.
\end{aligned}$$

Queries Phase 2: Similarly, when running the adversary $\mathcal{A}$, the relevant queries can occur according to the algorithm $\mathcal{B}_{A\_TIBGS}$ of Definition 5.3. The algorithm $\mathcal{B}$ answers these in the following way:

- *Group-Setup queries*: Given the public parameters $TIBGK$ and the identity ID_g of the group, the algorithm $\mathcal{B}$ similarly constructs a group private key $gsk_{\{ID_g\}} = \{y_0,\ y_1\} = \{g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu} \cdot \frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}},\ \ (g_3^{-\frac{1}{\nu}} \cdot g^{r_2})^{\frac{1}{H(ID_g)}}\}$ to the adversary $\mathcal{A}$. Setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$, $gsk_{\{ID_g\}} = \{y_0,\ y_1\} = \{g_4^b \cdot \mu^{r_2' \cdot H(ID_g)} \cdot \tau^{r_2'},\ \ g^{r_2'}\}$ is a valid private key, where we assure that $\nu \cdot H(ID_g) \neq 0$ **mod** q.
- *Join-User queries*: Given the public parameters $TIBGK$, the identity ID_g of the group and the identity ID_i of the group member (where $ID_g \neq ID_g^*$ and $ID_i \notin \{ID_0^*, ID_1^*\}$), the algorithm chooses random $r_2, r_3, r_4 \in \mathbb{Z}_q$ and computes

$$\begin{aligned}
v_0 &= g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu} \cdot \frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}} \cdot \vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3} \cdot \varpi^{r_4},\\
v_1 &= e(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g),\\
v_2 &= g^{r_4},\ \ v_3 = \left(g_3^{-\frac{1}{\nu}} \cdot g^{r_2}\right)^{\frac{1}{H(ID_g)}},\ \ v_4 = g^{r_3}.
\end{aligned}$$

  Finally, the algorithm outputs a member private key $usk_{\{ID_i\}} = \{v_0, v_1, v_2, v_3, v_4\}$ to the adversary $\mathcal{A}$. Similarly, setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$, $usk_{\{ID_i\}} = \{v_0, v_1, v_2, v_3, v_4\} = \{g_4^b \cdot \mu^{r_2' \cdot H(ID_g)} \cdot \tau^{r_2'} \cdot \vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3} \cdot \varpi^{r_4},\ \ e(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g),\ \ g^{r_4},\ \ g^{r_2'},\ \ g^{r_3}\}$ is a valid private key, where we assure that $\nu \cdot H(ID_g) \neq 0$ **mod** q.

Remark B.7. Where we maximize the adversary's advantage, thus v_4 is also passed to $\mathcal{A}$.

- *Revoke-User*: Given the public parameters $TIBGK$, the identity ID_i of the revoked group member and the revocation list RL^t_{ID} of the last duration t ($RL^t_{ID} = \varnothing$ when $t = 0$ and $\mathcal{A}$ did not query *Revoke-User* on inputs ID_g^*, ID_0^* (and ID_1^*)), the algorithm chooses random $r_1, r_3, r_4 \in \mathbb{Z}_q$ and computes

$$\begin{aligned}
T &= e(\vartheta^{H(ID_i)} \cdot \psi,\ \left(g_1^{-\frac{1}{\ell}} \cdot g^{r_1}\right)^{\frac{1}{H(ID_i)}}) \cdot e(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g),\\
v_2 &= g^{r_4}.
\end{aligned}$$

Finally, the algorithm outputs and adds a tuple $[ID_i, T, v_2]$ to the revocation list RL_{ID}^t, and then an updated revocation list RL_{ID}^{t+1} is published to the adversary $\mathcal{A}$. Similarly, setting $r_1' = (r_1 - \frac{a}{\ell}) \cdot \frac{1}{H(ID_i)}$,

$$\begin{aligned} T &= e\left(\vartheta^{H(ID_i)} \cdot \psi,\ \left(g_1^{-\frac{1}{\ell}} \cdot g^{r_1}\right)^{\frac{1}{H(ID_i)}}\right) \cdot e\left(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g\right) \\ &= e\left(\vartheta^{H(ID_i)} \cdot \psi,\ g^{\left(r_1 - \frac{a}{\ell}\right) \cdot \frac{1}{H(ID_i)}}\right) \cdot e\left(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g\right) \\ &= e\left(\vartheta^{H(ID_i)} \cdot \psi, g^{r_1'}\right) \cdot e\left(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g\right) \\ &= e\left(\vartheta^{(r_1'+r_3) \cdot H(ID_i)} \cdot \psi^{(r_1'+r_3)}, g\right), \end{aligned}$$

thus the tuple $[ID_i, T, v_2]$ is a valid data, where we assure that $\ell \cdot H(ID_i) \neq 0 \ \mathbf{mod}\ q$.

- *Sign queries*: Given the public parameters $TIBGK$, the identity ID_g of the group, the identity ID_i of the group member and the message $\mathfrak{M}$, the algorithm chooses random $r_2, r_3, r_4, r_5 \in \mathbb{Z}_q$ and computes

$$\begin{aligned} &\sigma_0 = g_3^{-\frac{1}{\nu}} \cdot \mu^{r_2} \cdot g_3^{-\frac{\lambda}{\nu} \cdot \frac{1}{H(ID_g)}} \cdot \tau^{\frac{r_2}{H(ID_g)}} \cdot \vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3} \cdot \varpi^{r_4} \cdot g_1^{-\frac{1}{\alpha}} \cdot \chi^{r_5} \cdot g_1^{-\frac{\pi}{\alpha} \cdot \frac{1}{H(\mathfrak{M})}} \cdot \kappa^{\frac{r_5}{H(\mathfrak{M})}}, \\ &\sigma_1 = e(\vartheta^{r_3 \cdot H(ID_i)} \cdot \psi^{r_3}, g), \\ &\sigma_2 = g^{r_4}, \\ &\sigma_3 = \left(g_3^{-\frac{1}{\nu}} \cdot g^{r_2}\right)^{\frac{1}{H(ID_g)}}, \\ &\sigma_4 = \left(g_1^{-\frac{1}{\alpha}} \cdot g^{r_5}\right)^{\frac{1}{H(\mathfrak{M})}}, \\ &\sigma_5 = g^{r_3}. \end{aligned}$$

Finally, the algorithm outputs a signature $\Phi = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$ to the adversary $\mathcal{A}$. Similarly, we maximize the adversary's advantage, thus σ_5 is also passed to the adversary $\mathcal{A}$. Setting $r_2' = (r_2 - \frac{b}{\nu}) \cdot \frac{1}{H(ID_g)}$ and $r_5' = (r_5 - \frac{a}{\alpha}) \cdot \frac{1}{H(\mathfrak{M})}$, $\Phi = \{\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5\}$ is a valid signature, where we assure that $\nu \cdot H(ID_g) \neq 0 \ \mathbf{mod}\ q$ and $\alpha \cdot H(\mathfrak{M}) \neq 0 \ \mathbf{mod}\ q$.

Guess: if the algorithm $\mathcal{B}$ does not abort as a consequence of one of the queries above, the adversary $\mathcal{A}$ will, with probability at least ε ($\varepsilon \geq \frac{1}{2}$) output a bit $x' \in \{0, 1\}$ and succeed ($x' = x$). We assume that $\Phi^{*'} = \{\sigma_0^{*'}, \sigma_1^{*'}, \sigma_2^{*'}, \sigma_3^{*'}, \sigma_4^{*'}, \sigma_5^{*'}\}$, where

$$\begin{aligned} \sigma_0^{*'} &= g_2^a \cdot g_4^b \cdot \vartheta^{r_2^* \cdot H(ID_{x'}^*)} \cdot \psi^{r_2^*} \cdot \mu^{r_3^* \cdot H(ID_g^*)} \cdot \tau^{r_3^*} \cdot \varpi^{r_4^*} \cdot \chi^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot \kappa^{r_5^*}, \\ \sigma_1^{*'} &= e(\vartheta^{r_2^* \cdot H(ID_{x'}^*)} \cdot \psi^{r_2^*}, g), \\ \sigma_2^{*'} &= g^{r_4^*}, \\ \sigma_3^{*'} &= g^{r_3^*}, \\ \sigma_4^{*'} &= g^{r_5^*}, \\ \sigma_5^{*'} &= g^{r_2^*}. \end{aligned}$$

So, compared with $\Phi^* = \{\sigma_0^*, \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*\} \leftarrow Sign(TIBGK,\ sk_{ID_x^*},\ \mathfrak{M}^*)$ in the Queries Phase 1, if $x' = x$, we can get the followings:

If $\ell \cdot H(ID_{x'}^*) \neq 0 \ \mathbf{mod}\ q$, or $\nu \cdot H(ID_g^*) \neq 0 \ \mathbf{mod}\ q$ or $\alpha \cdot H(\mathfrak{M}^*) \neq 0 \ \mathbf{mod}\ q$, then the algorithm $\mathcal{B}$ will abort.

If $\ell \cdot H(ID^*_{x'}) = 0 \textbf{ mod } q$, and $\nu \cdot H(ID^*_g) = 0 \textbf{ mod } q$ and $\alpha \cdot H(\mathfrak{M}^*) = 0 \textbf{ mod } q$, then the algorithm $\mathcal{B}$ computes and outputs

$$\begin{aligned}
&\sqrt[2]{\frac{\sigma_0^{*'}}{g_1^{-x_1} \cdot g_3^{-x_2} \cdot g^{r_2^* \cdot H(ID^*_{x'})} \cdot g^{r_2^* \cdot \partial} \cdot g^{r_3^* \cdot H(ID^*_g)} \cdot g^{r_3^* \cdot \lambda} \cdot g^{r_4^* \cdot \eta} \cdot g^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot g^{r_5^* \cdot \pi}}} \\
&= \sqrt[2]{\frac{g_2^a \cdot g_4^b \cdot \vartheta^{r_2^* \cdot H(ID^*_{x'})} \cdot \psi^{r_2^*} \cdot \mu^{r_3^* \cdot H(ID^*_g)} \cdot \tau^{r_3^*} \cdot \varpi^{r_4^*} \cdot \chi^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot \kappa^{r_5^*}}{g_1^{-x_1} \cdot g_3^{-x_2} \cdot g^{r_2^* \cdot H(ID^*_{x'})} \cdot g^{r_2^* \cdot \partial} \cdot g^{r_3^* \cdot H(ID^*_g)} \cdot g^{r_3^* \cdot \lambda} \cdot g^{r_4^* \cdot \eta} \cdot g^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot g^{r_5^* \cdot \pi}}} \\
&= \sqrt[2]{\frac{(g^b \cdot g^{-x_1})^a \cdot (g^a \cdot g^{-x_2})^b \cdot (g_2^\ell \cdot g)^{r_2^* H(ID^*_{x'})} \cdot (g^\partial)^{r_2^*} \cdot (g_4^\nu \cdot g)^{r_3^* H(ID^*_g)} \cdot (g^\lambda)^{r_3^*} \cdot (g^\eta)^{r_4^*} \cdot (g_2^\alpha \cdot g)^{r_5^* H(\mathfrak{M}^*)} \cdot (g^\pi)^{r_5^*}}{g_1^{-x_1} \cdot g_3^{-x_2} \cdot g^{r_2^* \cdot H(ID^*_{x'})} \cdot g^{r_2^* \cdot \partial} \cdot g^{r_3^* \cdot H(ID^*_g)} \cdot g^{r_3^* \cdot \lambda} \cdot g^{r_4^* \cdot \eta} \cdot g^{r_5^* \cdot H(\mathfrak{M}^*)} \cdot g^{r_5^* \cdot \pi}}} \\
&= g^{a \cdot b},
\end{aligned}$$

which is the solution to the given CDH problem.

Now, we analyze the probability of the algorithm $\mathcal{B}$ not aborting. For the simulation to complete without aborting, we require that all *Group-Setup* queries will have $\nu \cdot H(ID_g) \neq 0 \textbf{ mod } q$, all *Join-User* queries will have $\nu \cdot H(ID_g) \neq 0 \textbf{ mod } q$, all *Revoke-User* queries will have $\ell \cdot H(ID_i) \neq 0 \textbf{ mod } q$, and all *Sign* queries will have $\nu \cdot H(ID_g) \neq 0 \textbf{ mod } q$ and $\alpha \cdot H(\mathfrak{M}) \neq 0 \textbf{ mod } q$ in the queries Phase 1 and 2, and that $\ell \cdot H(ID^*_{x'}) = 0 \textbf{ mod } q$, and $\nu \cdot H(ID^*_g) = 0 \textbf{ mod } q$ and $\alpha \cdot H(\mathfrak{M}^*) = 0 \textbf{ mod } q$ in guess. If the algorithm $\mathcal{B}$ does not abort, then the following conditions must hold:

(a) $\nu \cdot H(ID_{g_i}) \neq 0 \textbf{ mod } q$ in *Group-Setup* queries, with $i = 1, 2 \ldots q_{g_1}$;
(b) $\nu \cdot H(ID_{g_i}) \neq 0 \textbf{ mod } q$ in *Join-User* queries, with $i = 1, 2 \ldots q_{j_1}$;
(c) $\ell \cdot H(ID_i) \neq 0 \textbf{ mod } q$ in *Revoke-User* queries, with $i = 1, 2 \ldots q_{r_1}$;
(d) $\nu \cdot H(ID_{g_i}) \neq 0 \textbf{ mod } q$ and $\alpha \cdot H(\mathfrak{M}_i) \neq 0 \textbf{ mod } q$ in *Sign* queries, with $i = 1, 2 \ldots q_{s_1}$;
(e) $\nu \cdot H(ID_{g_i}) \neq 0 \textbf{ mod } q$ in *Group-Setup* queries, with $i = 1, 2 \ldots q_{g_2}$;
(f) $\nu \cdot H(ID_{g_i}) \neq 0 \textbf{ mod } q$ in *Join-User* queries, with $i = 1, 2 \ldots q_{j_2}$;
(g) $\ell \cdot H(ID_i) \neq 0 \textbf{ mod } q$ in *Revoke-User* queries, with $i = 1, 2 \ldots q_{r_2}$;
(h) $\nu \cdot H(ID_{g_i}) \neq 0 \textbf{ mod } q$ and $\alpha \cdot H(\mathfrak{M}_i) \neq 0 \textbf{ mod } q$ in *Sign* queries, with $i = 1, 2 \ldots q_{s_2}$;
(i) the algorithm $\mathcal{B}$ does not abort in guess, namely $\ell \cdot H(ID^*_{x'}) = 0 \textbf{ mod } q$, and $\nu \cdot H(ID^*_g) = 0 \textbf{ mod } q$ and $\alpha \cdot H(\mathfrak{M}^*) = 0 \textbf{ mod } q$.

Then we will define the events D_{1_i}, F_{1_i}, E_{1_i}, T_{1_i}, L_{1_i}, D_{2_i}, F_{2_i}, E_{2_i}, T_{2_i}, L_{2_i}, R^*, F^*, S^* as

D_{1_i}: $\nu \cdot H(ID_{g_i}) \neq 0 \textbf{ mod } q$, with $i = 1, 2 \ldots q_{g_1}$;
F_{1_i}: $\nu \cdot H(ID_{g_i}) \neq 0 \textbf{ mod } q$, with $i = 1, 2 \ldots q_{j_1}$;
E_{1_i}: $\ell \cdot H(ID_i) \neq 0 \textbf{ mod } q$, with $i = 1, 2 \ldots q_{r_1}$;
T_{1_i}: $\nu \cdot H(ID_{g_i}) \neq 0 \textbf{ mod } q$, with $i = 1, 2 \ldots q_{s_1}$;
L_{1_i}: $\alpha \cdot H(\mathfrak{M}_i) \neq 0 \textbf{ mod } q$, with $i = 1, 2 \ldots q_{s_1}$;
D_{2_i}: $\nu \cdot H(ID_{g_i}) \neq 0 \textbf{ mod } q$, with $i = 1, 2 \ldots q_{g_2}$;
F_{2_i}: $\nu \cdot H(ID_{g_i}) \neq 0 \textbf{ mod } q$, with $i = 1, 2 \ldots q_{j_2}$;
E_{2_i}: $\ell \cdot H(ID_i) \neq 0 \textbf{ mod } q$, with $i = 1, 2 \ldots q_{r_2}$;
T_{2_i}: $\nu \cdot H(ID_{g_i}) \neq 0 \textbf{ mod } q$, with $i = 1, 2 \ldots q_{s_2}$;
L_{2_i}: $\alpha \cdot H(\mathfrak{M}_i) \neq 0 \textbf{ mod } q$, with $i = 1, 2 \ldots q_{s_2}$;
R^*: $\ell \cdot H(ID^*_{x'}) = 0 \textbf{ mod } q$;
F^*: $\nu \cdot H(ID^*_g) = 0 \textbf{ mod } q$;
S^*: $\alpha \cdot H(\mathfrak{M}^*) = 0 \textbf{ mod } q$.

The probability of $\mathcal{B}$ not aborting is

$$\Pr(not\_abort) = \Pr\left(\bigcap_{i=1}^{q_{g_1}} D_{1_i} \wedge \bigcap_{i=1}^{q_{j_1}} F_{1_i} \wedge \bigcap_{i=1}^{q_{r_1}} E_{1_i} \wedge \bigcap_{i=1}^{q_{s_1}} (T_{1_i} \wedge L_{1_i}) \wedge \bigcap_{i=1}^{q_{g_2}} D_{2_i} \wedge \bigcap_{i=1}^{q_{j_2}} F_{2_i} \wedge \bigcap_{i=1}^{q_{r_2}} E_{2_i} \right.$$
$$\left. \wedge \bigcap_{i=1}^{q_{s_2}} (T_{2_i} \wedge L_{2_i}) \wedge R^* \wedge F^* \wedge S^* \right).$$

It is easy to see that the events $\bigcap_{i=1}^{q_{g_1}} D_{1_i}$, $\bigcap_{i=1}^{q_{j_1}} F_{1_i}$, $\bigcap_{i=1}^{q_{r_1}} E_{1_i}$, $\bigcap_{i=1}^{q_{s_1}} T_{1_i}$, $\bigcap_{i=1}^{q_{s_1}} L_{1_i}$, $\bigcap_{i=1}^{q_{g_2}} D_{2_i}$, $\bigcap_{i=1}^{q_{j_2}} F_{2_i}$, $\bigcap_{i=1}^{q_{r_2}} E_{2_i}$, $\bigcap_{i=1}^{q_{s_2}} T_{2_i}$, $\bigcap_{i=1}^{q_{s_2}} L_{2_i}$, R^*, F^* and S^* are independent. Then we may compute

$$\Pr\left(\bigcap_{i=1}^{q_{g_1}} D_{1_i}\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_{g_1}} \neg D_{1_i}\right) = 1 - q_{g_1} \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_{g_1}}{q};$$

$$\Pr\left(\bigcap_{i=1}^{q_{g_2}} D_{2_i}\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_{g_2}} \neg D_{2_i}\right) = 1 - q_{g_2} \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_{g_2}}{q};$$

$$\Pr\left(\bigcap_{i=1}^{q_{j_1}} F_{1_i}\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_{j_1}} \neg F_{1_i}\right) = 1 - q_{j_1} \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_{j_1}}{q};$$

$$\Pr\left(\bigcap_{i=1}^{q_{j_2}} F_{2_i}\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_{j_2}} \neg F_{2_i}\right) = 1 - q_{j_2} \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_{j_2}}{q};$$

$$\Pr\left(\bigcap_{i=1}^{q_{r_1}} E_{1_i}\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_{r_1}} \neg E_{1_i}\right) = 1 - q_{r_1} \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_{r_1}}{q};$$

$$\Pr\left(\bigcap_{i=1}^{q_{r_2}} E_{2_i}\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_{r_2}} \neg E_{2_i}\right) = 1 - q_{r_2} \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_{r_2}}{q};$$

$$\Pr\left(\bigcap_{i=1}^{q_{s_1}} T_{1_i}\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_{s_1}} \neg T_{1_i}\right) = 1 - q_{s_1} \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_{s_1}}{q};$$

$$\Pr\left(\bigcap_{i=1}^{q_{s_2}} T_{2_i}\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_{s_2}} \neg T_{2_i}\right) = 1 - q_{s_2} \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_{s_2}}{q};$$

$$\Pr\left(\bigcap_{i=1}^{q_{s_1}} L_{1_i}\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_{s_1}} \neg L_{1_i}\right) = 1 - q_{s_1} \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_{s_1}}{q};$$

$$\Pr\left(\bigcap_{i=1}^{q_{s_2}} L_{2_i}\right) = 1 - \Pr\left(\bigcup_{i=1}^{q_{s_2}} \neg L_{2_i}\right) = 1 - q_{s_2} \cdot \frac{1^k}{1^k \cdot q} = 1 - \frac{q_{s_2}}{q};$$

$$\Pr(R^*) = \frac{1^k}{1^k \cdot q} = \frac{1}{q}; \quad \Pr(F^*) = \frac{1^k}{1^k \cdot q} = \frac{1}{q}; \quad \Pr(S^*) = \frac{1^k}{1^k \cdot q} = \frac{1}{q} \cdot$$

So,

$$\begin{aligned}\Pr(not\_abort) &= \Pr\Bigg(\bigcap_{i=1}^{q_{g_1}} D_{1_i} \wedge \bigcap_{i=1}^{q_{j_1}} F_{1_i} \wedge \bigcap_{i=1}^{q_{r_1}} E_{1_i} \wedge \bigcap_{i=1}^{q_{s_1}} (T_{1_i} \wedge L_{1_i}) \wedge \bigcap_{i=1}^{q_{g_2}} D_{2_i} \wedge \bigcap_{i=1}^{q_{j_2}} F_{2_i} \wedge \bigcap_{i=1}^{q_{r_2}} E_{2_i} \\ &\quad \wedge \bigcap_{i=1}^{q_{s_2}} (T_{2_i} \wedge L_{2_i}) \wedge R^* \wedge F^* \wedge S^*\Bigg) \\ &= \left(1-\frac{q_{g_1}}{q}\right)\cdot\left(1-\frac{q_{j_1}}{q}\right)\cdot\left(1-\frac{q_{r_1}}{q}\right)\cdot\left(1-\frac{q_{s_1}}{q}\right)^2\cdot\left(1-\frac{q_{g_2}}{q}\right)\cdot\left(1-\frac{q_{j_2}}{q}\right)\cdot\left(1-\frac{q_{r_2}}{q}\right)\cdot\left(1-\frac{q_{s_2}}{q}\right)^2\cdot\frac{1}{q^3}.\end{aligned}$$

Then we can get that

$$\varepsilon' = \left(1-\frac{q_{g_1}}{q}\right)\cdot\left(1-\frac{q_{j_1}}{q}\right)\cdot\left(1-\frac{q_{r_1}}{q}\right)\cdot\left(1-\frac{q_{s_1}}{q}\right)^2\cdot\left(1-\frac{q_{g_2}}{q}\right)\cdot\left(1-\frac{q_{j_2}}{q}\right)\cdot\left(1-\frac{q_{r_2}}{q}\right)\cdot\left(1-\frac{q_{s_2}}{q}\right)^2\cdot\frac{\varepsilon-\frac{1}{2}}{q^3}.$$

If the simulation does not abort, the adversary $\mathcal{A}$ will break the anonymity with probability at least $\varepsilon - \frac{1}{2}$. The algorithm $\mathcal{B}$ can then compute $g^{a\cdot b}$ from the forgery as shown above. The time complexity of the algorithm $\mathcal{B}$ is dominated by the time for the exponentiations and multiplications in the queries. We assume that the time for integer addition and integer multiplication, and the time for hash computation can both be ignored, then the time complexity of the algorithm $\mathcal{B}$ is

$$\begin{aligned}\hbar' = \hbar + O((q_{g_1}+q_{g_2})\cdot(5\cdot C_{exp}+4\cdot C_{mul_1}) + (q_{j_1}+q_{j_2})\cdot(10\cdot C_{exp}+7\cdot C_{mul_1}+1\cdot C_{pair}) \\ + (q_{r_1}+q_{r_2})\cdot(6\cdot C_{exp}+3\cdot C_{mul_1}+2\cdot C_{pair}+C_{mul_2}) + (q_{s_1}+q_{s_2})\cdot(15\cdot C_{exp}+12\cdot C_{mul_1}+1\cdot C_{pair})).\end{aligned}$$

Thus, Theorem 7.3 follows. □

Acknowledgements. This work is supported by the National Natural Science Foundation of China (Nos. 61402055, 61462048, 61504013), the Natural Science Foundation of Hunan Province (No. 2016JJ3012), and the Scientific Research Project of Hunan Provincial Education Department (No. 15C0041, 15A007, 15C0779).

References

[1] M.H. Au, J.K. Liu, T.H. Yuen and D.S. Wong, ID-based ring signature scheme secure in the standard mode, In *Proc. of IWSEC* (2006) 1–16.

[2] M.H. Au, J.K. Liu, W. Susilo and T.H. Yuen, Secure ID-Based Linkable and Revocable-iff-Linked Ring Signature with Constant-Size Construction. *Theoret. Comput. Sci.* **469** (2013) 1–14.

[3] G. Ateniese, J. Camenisch, M. Joye and G. Tsudik, A practical and provably secure coalition-resistant group signature scheme. In Vol. 1880 of *Lect. Notes Comput. Sci.* Springer (2000) 255–270.

[4] G. Ateniese, D. Song and G. Tsudik, Quasi-Efficient Revocation in Group Signatures. In *Financial Cryptography'02*. Vol. 2357 of *Lect. Notes Comput. Sci.* Springer (2002) 183–197.

[5] P.S.L.M. Barreto, B. Libert, N. McCullagh and J. Quisquater, Efficient and Provably-Secure Identity-Based Signatures and Signcryption from Bilinear Maps. In *Asiacrypt 2005*, edited by B. Roy. Vol. 3788 of *Lect. Notes Comput. Sci.* Springer-Verlag, Berlin (2005) 515–532.

[6] M. Bellare, D. Micciancio and B. Warinschi, Foundations of group signatures: Formal definitions, simplified require-ments, and a construction based on general assumptions. In *Eurocrypt'03*. Vol. 2656 of *Lect. Notes Comput. Sci.* Springer (2003) 614–629.

[7] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing. In *Advances in Cryptology-CRYPTO 2001*, edited by J. Kilian. Vol. 2139 of *Lect. Notes Comput. Sci.* Springer-Verlag, Berlin (2001) 213–229.

[8] D. Boneh and M. Hanburg, Generalized identity based and broadcast encryption schemes. In *Advances in Cryptology-ASIACRYPT 2008*, edited by J. Pieprzyk. Vol. 5350 of *Lect. Notes Comput. Sci.* Springer-Verlag, Berlin (2008) 455–470.

[9] D. Boneh and H. Shacham, Group signatures with verifier-local revocation. In *ACM-CCS'04* (2004) 168–177.

[10] D. Boneh, X. Boyen and H. Shacham, Short Group Signatures. In *Crypto'04*. Vol. 3152 of *Lect. Notes Comput. Sci.* Springer (2004) 41–55.

[11] E. Bresson and J. Stern, Efficient Revocation in Group Signatures. In *PKC'01*. Vol. 1992 of *Lect. Notes Comput. Sci.* Springer (2001) 190–206.

[12] E. Brickell, An efficient protocol for anonymously providing assurance of the container of the private key. Sub-mission to the Trusted Computing Group (2003).

[13] E. Brickell, J. Camenisch and L. Chen, Direct Anonymous Attestation. In *ACM-CCS'04* (2004) 132–145.

[14] J. Camenisch and A. Lysyanskaya, Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *Crypto'02*. Vol. 2442 of *Lect. Notes Comput. Sci.* Springer (2002) 61–76.

[15] J. Camenisch, M. Kohlweiss and C. Soriente, An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. In *PKC'09*. Vol. 5443 of *Lect. Notes Comput. Sci.* Springer (2009) 481–500.

[16] J.C. Cha and J.H. Cheon, An identity-based signature from gap Diffie–Hellman groups. In *Public Key Cryptography – PKC 2003*, edited by Y. Desmedt. Vol. 2567 of *Lect. Notes Comput. Sci.* Springer-Verlag, Berlin (2002) 18–30.

[17] D. Chaum and E. van Heyst, Group Signatures. In *Eurocrypt'91*. Vol. 547 of *Lect. Notes Comput. Sci.* Springer (1991) 257–265.

[18] K. Emura, A. Miyaji and K. Omote, An r-Hiding Revocable Group Signature Scheme: Group Signatures with the Property of Hiding the Number of Revoked Users. *J. Appl. Math.* **2014** (2014) 14.

[19] F. Hess, Efficient identity based signature schemes based on pairings. In *Selected Areas in Cryptography 9th Annual International Workshop, SAC 2002*, edited by K. Nyberg, H. Heys. Vol. 2595 of *Lect. Notes Comput. Sci.* Springer-Verlag, Berlin (2003) 310–324.

[20] L. Ibraimi, S. Nikova, P. Hartel and W. Jonker, An Identity-Based Group Signature with Membership Revocation in the Standard Model, available at: http:/doc.utwente.nl/72270/1/Paper.pdf.

[21] B. Libert and D. Vergnaud, Group Signatures with Verifier-Local Revocation and Backward Unlinkability in the Standard Model. In *CANS'09*. Vol. 5888 of *Lect. Notes Comput. Sci.* Springer (2009) 498–517.

[22] B. Libert, T. Peters and M. Yung, Scalable Group Signatures with Revocation. *Advances in Cryptology-EUROCRYPT 2012*. Vol. 7323 of *Lect. Notes Comput. Sci.* Springer-Verlag (2012) 609-627.

[23] B. Libert, T. Peters and M. Yung, Scalable Group Signatures with Almost-for-Free Revocation. *Advances in Cryptology-CRYPTO 2012*. Vol. 7417 of *Lect. Notes Comput. Sci.* Springer-Verlag (2012) 571–589.

[24] T. Nakanishi and N. Funabiki, Verifier-Local Revocation Group Signature Schemes with Backward Unlinkability from Bilinear Maps. In *Asiacrypt'05*. Vol. 5443 of *Lect. Notes Comput. Sci.* Springer (2009) 533-548.

[25] T. Nakanishi, H. Fujii, Y. Hira and N. Funabiki, Revocable Group Signature Schemes with Constant Costs for Signing and Verifying. In *PKC'09*. Vol. 5443 of *Lect. Notes Comput. Sci.* Springer (2009) 463–480.

[26] L. Nguyen, Accumulators from Bilinear Pairings and Applications. In *CT-RSA'05*. Vol. 3376 of *Lect. Notes Comput. Sci.* Springer (2005) 275–292.

[27] K.G. Paterson and J.C.N. Schuldt, Efficient identity-based signatures secure in the standard model. In *ACISP 2006*. Vol. 4058 of *Lect. Notes Comput. Sci.* Springer-Verlag (2006) 207–222.

[28] H. Singh and G.K. Verma, ID-based proxy signature scheme with message recovery. *J. Systems Software* **85** (2012) 209–214.

[29] B. Waters, Efficient identity-based encryption without random oracles, *Advances in Cryptology-EUROCRYPT 2005*. Vol. 3494 of *Lect. Notes Comput. Sci.* Springer-Verlag (2005) 114–127.

[30] F.T. Wen, S.J. Cui and J.N. Cui, An ID-based Proxy Signature Scheme Secure Against Proxy Key Exposure. *Int. J. Adv. Comput. Technol.* **3** (2011) 108–116.

[31] W. Wu, Y. Mu, W. Susilo, J. Seberry and X.Y. Huang, Identity-Based Proxy Signature from Pairings, In *ATC 2007*, edited by B. Xiao *et al.* Vol. 4610 of *Lect. Notes Comput. Sci.* Springer-Verlag, Berlin (2007) 22–31.

[32] F. Zhang and K. Kim, ID-based blind signature and ring signature from pairings. in *Asiacrypt 2002*. Vol. 2501 *Lect. Notes Comput. Sci.* Springer-Verlag, Berlin (2002) 533–547.

[33] S. Zhou, D. Lin, Shorter Verifier-Local Revocation Group Signatures from Bilinear Maps. In *CANS'06*. Vol. 4301 of *Lect. Notes Comput. Sci.* Springer (2006) 126–143.

Communicated by S. Mesnager.
Received June 19, 2016. Accepted September 19, 2016.