An Identity-Based Group Signature with Membership Revocation in the Standard Model

Luan Ibraimi[1], Svetla Nikova[1,3], Pieter Hartel[1], Willem Jonker[1,2]

[1] EWI, University of Twente, the Netherlands
[2] Philips Research, the Netherlands
[3] ESAT/COSIC, Katholieke Universiteit Leuven, Belgium

Abstract. Group signatures allow group members to sign an arbitrary number of messages on behalf of the group without revealing their identity. Under certain circumstances the group manager holding a tracing key can reveal the identity of the signer from the signature. Practical group signature schemes should support membership revocation where the revoked member loses the capability to sign a message on behalf of the group without influencing the other non-revoked members. A model known as *verifier-local revocation* supports membership revocation. In this model the trusted revocation authority sends revocation messages to the verifiers and there is no need for the trusted revocation authority to contact non-revoked members to update their secret keys. Previous constructions of verifier-local revocation group signature schemes either have a security proof in the random oracle model or are non-identity based. A security proof in the random oracle model is only a heuristic proof and non-identity-based group signature suffer from standard Public Key Infrastructure (PKI) problems, i.e. the group public key is not derived from the group identity and therefore has to be certified.
In this work we construct the first verifier-local revocation group signature scheme which is identity-based and which has a security proof in the standard model. In particular, we give a formal security model for the proposed scheme and prove that the scheme has the property of anonymity under the decision Linear (DLIN) assumption and it is fully-traceable under the Computation Diffie-Hellman (CDH) assumption. The proposed scheme is based on prime order bilinear groups.

1 Introduction

In public key cryptography, the authenticity of cryptographic keys is important. The party who encrypts the data (in case of a public key encryption), or a party who verifies a signature (in case of a digital signature), needs to be assured that the public key belongs to the right user who is also in possession of the corresponding private key. In a Public Key Infrastructure (PKI), the Certificate Authority (CA) generates a digital certificate, which contains a digital signature, to assure that the public key belongs to the right user. Whenever a user wants to use a public key, the user has to obtain the digital certificate and verify the signature. In practice PKI technology suffers from many drawbacks such as certificate verification, revocation, distribution, storage, etc [17]. On the other hand, in the Identity-Based Cryptography, introduced by Shamir [21], the public key is derived from the identity of the user (e.g.name, email address, IP address), thus there is no need for a use of digital certificates to certify the public key.

The aim of this paper is to construct an identity-based group signature scheme which supports member revocations and which has a security proof in the standard model. Group signatures, introduced by Chaum and Van Heyst [14], allow a group member to sign a message on behalf of the group such that other group members cannot reveal the identity of the signer, but in certain circumstances the group manager has the power to reveal the identity of the signer from the signature. The verifier of the group signature uses the public key of

the group to verify that the signature is generated by a group member. Non-identity based group signatures suffer from the aforementioned PKI problems, while by using an identity-based group signature one avoids the need to use digital certificates. In addition to removing the need for digital certificates, supporting membership revocation is important as well. In particular, there are situations when a group member may leave the group voluntarily or a group member might get compromised. Previous group signatures in the literature, which support membership revocation, have either a security proof in the heuristic random oracle model or are non-identity based.

1.1 Our Contribution

In this paper we propose a group signature scheme, named as verifier-local revocation identity-based group signature (VLR-IBGS), which simultaneously satisfies the following desirable properties:

1. VLR-IBGS supports membership revocation such that a group member losses his signing capabilities after the revocation. In general, revocation may happen when a group member leaves the group voluntarily, when the member secret key is compromised, or when the member is misbehaving by giving his secret key to unauthorized users.
2. VLR-IBGS has a security proof in the standard model. In particular, we show that the scheme has the property of anonymity under the Decisional Linear (DLIN) assumption and is fully-traceable under the Computational Diffie-Hellman (CDH) assumption. Anonymity ensures that the digital signature does not reveal the identity of the signer while the owner of a secret key can detect whether the signature was created by her secret key and the full traceability allows the group manager to recover the identity of the signer whenever a dispute arises.
3. VLR-IBGS is identity-based where the group public key is derived from the group identity and does not have to be certified.

We believe that the design of a group signature scheme which satisfies the above properties, is interesting for two reasons. The first reason is that VLR-IBGS fills the gap with existing group signature by providing a more comprehensive scheme with more interesting properties and the other reason is that VLR-IBGS makes group signatures even more useful for constructing other primitives, specifically for constructing sanitizable signatures. Sanitizable signatures allow a semi-trusted party, called the sanitizer, to modify parts of the signed data without interacting with the original signer. Berzuska et. al. [9], give the first sanitizable signature which uses group signature as a building block. However, when non-identity based group signatures are used to construct sanitazable signatures, including the scheme of Berzuska et al. [9], the public key of the original signer and the sanitizer needs to be registered (i.e. certified). In this context, a group signature with the above properties removes the need to certify the public key and also allows the original signer to revoke the sanitizer, if required.

Our contribution can be viewed as complementing the work of Smart and Warinschi [22] and Libert and Vergnaud [19]. Smart and Warinschi [22] provide a model for an identity-based group signature scheme and give a generic construction based on the hierarchical identity-based encryption (HIBE) [15] scheme and the Boyen and Waters [7,8] signature schemes. The main difference between our work and the work of [22] is that the latter focuses merely on constructing an identity-based group signature scheme, whereas our work focuses on constructing an identity-based group signature scheme which supports membership revocation.

Libert and Vergnaud [19] give an non-identity based group signature which supports membership revocation and which is secure in the standard model. Computationally our scheme is more efficient than Libert and Vergnaud scheme, since the latter uses pairing operations when the signature is created, however our scheme uses pairing operations only in the verification phase. It is also important to mention that the security proof of the Libert and Vergnaud scheme is based on slightly stronger assumptions than the security proof of our scheme. On the other hand the Libert-Vergnaud scheme support backward unlinkability which is used to protect anonymity of signatures created by revoked members at the time when the users were not revoked; whereas our scheme does not support this property.

1.2 More Related Work

Group Signatures. Since Chaum and Van Heyst [14] introduced the concept, a number of group signatures schemes have been proposed [1,2,3,4,10,11,12,18,23,**?**]. Many efficient group signature scheme have been proposed in the random oracle model, however the random oracle model is a heuristic security model. Canetti et al. [13] shows that there are signature schemes which are secure in the random oracle model but which are insecure for any implementation of the random oracle. Bellare et. al. [3] proposed security definitions for group signature schemes and gave the first construction provable secure in the standard model. Boyen and Waters [7,8] suggested an efficient group signature with security proofs in the standard model. The Boyen and Waters construction is a two-level signature scheme in which the first level of the signature is the signers identity and the second level is the message to be signed. In a later scheme of Boyen and Waters [8], to hide the identity of the signer, bilinear groups of composite order are used, as well as non-interactive zero-knowledge (NIZK) proofs. The assumptions under which the scheme is proven secure imply that it is difficult to factor the composite order of the bilinear group. The scheme is inefficient compared to schemes which use prime order groups since it uses larger group elements with more expensive operations.

Groth [16] gives a practical group signature scheme based on prime order bilinear groups and a security proof under standard assumptions: the strong Diffie-Hellman assumption (q-SDH), the decision Linear (DLIN) assumption and the unfakeability assumption (q-U). The size of the q-SDH and q-U assumptions depends on the number of queries asked by the adversary. Therefore, the security proof under these assumptions requires larger security parameters compared to other security proof which use constant size assumptions where the size of the assumption does not depend on the number of queries asked by the adversary.

Verifier-Local Revocation Group Signatures. The simplest revocation method is due to Ateniese et. al. [2] where the group manager changes the group public key and the secret keys of non-revoked members. However the scheme is not efficient since the key update can be a bottleneck for both the group manager and non-revoked members. Another method [4,12] is to broadcast a small message to all signers and verifiers. Only non-revoked members can use the broadcast message to update their secret keys and generate a valid signature. For the revoked members the broadcast message is a redundant value and cannot help them to update their secret keys. The drawback of this approach is that the signer has to perform computations depending on the number of revoked members. The high number of signer computations makes this model unsuitable for low-cost devices.

A more efficient solution known as verifier-local revocation (VLR) [23,6,20] is to send revocation tokens to verifiers. In this model there is no need for the trusted revocation authority to contact non-revoked members to update their secret keys while the verifier performs com-

Reference	Membership Revocation	Identity-Based	Security Proof
Boyen-Waters [7,8]	No	Yes	Standard Model
Groth [16]	No	No	Standard Model
Boneh-Shacham [6]	Yes	No	Random Oracle Model
Smart-Warinschi [22]	No	Yes	Standard Model
Libert-Vergnaud [19]	Yes	No	Standard Model
This paper	Yes	Yes	Standard Model

Table 1. Comparison of our scheme with the most efficient related work

putations depending on the number of revoked members. The Song [23] scheme is based on the strong RSA assumption and it is inefficient due to the use of inefficient zero-knowledge proofs. The Boneh and Shacham [6] scheme is based on bilinear maps and has short signatures. Nakanishi and Funabiki [20] have proposed a VLR group signature scheme with the property of backward unlinkability. This property means that all signatures produced by the member before the revocation remain anonymous. The security proofs of the Boneh and Shacham [6] and Nakanishi and Funabiki [20] scheme is in the random oracle model.

In table 1 we compare our scheme with the most efficient previous work. The comparison is based on following properties: a) functionality of the scheme - whether the scheme supports membership revocation, b) the way of generating the group public key - whether the scheme is identity-based, and c) security proof - whether the scheme has a security proof in the standard model or a random oracle model.

Organization of the paper. In section 2 we define the syntax of VLR-IBGS scheme and the required security properties. In this section we also review the basics of bilinear pairing and complexity assumptions under which the security of the proposed scheme is based. In section 3 we present the construction of the scheme, its correctness proof along with the formal security proof. The last section concludes the paper.

Notation. If S is a set then $s \in_R S$ denotes that s is selected uniformly at random from S. If $\lambda \in \mathbb{N}$, then 1^λ denotes the string consisting of λ ones. $\mathcal{A}$ stands for the adversary which is a polynomial-time algorithm. We write $\mathcal{A}(x, y, ...,)$ to indicate that the algorithm $\mathcal{A}$ has inputs $x, y, ...,$, and we write $z \leftarrow \mathcal{A}(x, y, ...,)$ to indicate the operation of running $\mathcal{A}$ with inputs $x, y, ...$ and getting z as output. We write $\{S_i\}_{i=1}^n$ to denote $\{S_1, S_2, ..., S_n\}$. A function $P(k) : \mathbb{Z} \to \mathbb{R}$ is negligible if, for every polynomial $f(k)$, there exists an integer N_f such that $P(k) \leq \frac{1}{f(k)}$ for all $k \geq N_f$. Unless noted otherwise, all algorithms are randomized and run in polynomial time.

2 Model and Security Definitions

Definition 1. *The verifier-local revocation identity-based group signature scheme* VLR-IBGS *consists of five algorithms* (**Setup**, **Group Setup**, **Enroll**, **Sign**, **Verify**)*:*

- **Setup**(1^λ) : *run by TA, the algorithm produces the master public key mpk and the master secret key msk for the security parameter* $\lambda \in \mathbb{N}$*. The master public key mpk is stored in a publicly accessible database. We assume that all the other algorithms always include the master public key mpk as their input.*
- **Group Setup**(msk, G) : *run by TA, the algorithm produces a group secret key* sk_G*, which is given to a group manager.*

– **Enroll**(sk_G, U) : *run by a group manager, the algorithm produces a member secret key* $sk_{G,U}$ *which is given to a group member.*
– **Sign**$(M, sk_{G,U})$: *run by a group member, the algorithm produces a signature* σ *on the message* M*.*
– **Verify**$(M, \sigma, \Re, G)$: *run by a verifier, the algorithm returns true if* σ *is a valid signature i.e. the signature is issued by a signer who is in the group* G *and does not have a revocation token* $\mathcal{T}_{G,U}$ *in the list of revoked members* $\Re$*. Otherwise, the algorithm returns* $false$*.*

For correctness is required for all $sk_G \leftarrow \mathsf{Group\ Setup}(msk, G)$, all $sk_{G,U} \leftarrow \mathsf{Enroll}\ (sk_G, U)$, any message $M \in \{0,1\}^*$, if the signer U does not have a revocation token $\mathcal{T}_{G,U}$ in the list of revoked members $\Re$, then:

$$\Pr\left[\mathsf{Verify}\left(M, \mathsf{Sign}(M, \mathsf{Enroll}(sk_G, U)), \Re, G\right) = true\right] = 1$$

2.1 Definition of Security

The security requirements of a VLR-IBGS scheme must guarantee anonymity and full traceability.

The property of anonymity requires from a group signature scheme to provide anonymity for the signer. In particular, the signature should not reveal the identity of the signer and an adversary should not be able to distinguish a signature generated by member U_0 from a signature generated by member U_1. We formalize this property by considering a security game that involves a challenger and an adversary. First, the challenger runs the setup algorithm and generates a master public key mpk and a master secret key msk, and gives to the adversary the master public key mpk. Then, the adversary can adaptively submit four types of queries: group setup queries, enroll queries, signature queries and revocation queries. A group setup query consists of a group identity G, and the challenger answers the query by running the group setup algorithm on input of the master secret key and the group identity. An enroll query consists of a member identity and a group identity, and is answered by running the enroll algorithm on input of the group secret key and the member identity. A signing query consists of a message m, a group identity and a member identity, and the challenger answers the query by running the signature-generation algorithm on input of the message and the group member secret key. A revocation query consists of a member identity and a group identity, and the challenger answers the query by returning a revocation token for the member identity. At some point, the adversary sends to the challenger two member identities, a message and a group identity. The challenger chooses one member identity at random and generates a signature under the secret key of this identity. Finally, the adversary is successful if it correctly guesses which of the two member secret keys was used by the challenger to generate the signature. More formally, this property is captured by the following definition.

Definition 2. (Anonymity). *The VLR-IBGS scheme is said to fulfill the requirement of anonymity if any* $\mathcal{A}$ *has only a negligible advantage in the anonymity game which is defined as follows:*

– Setup. *The challenger runs* $(mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda)$ *and gives* mpk *to* $\mathcal{A}$*.*
– Query Phase 1. $\mathcal{A}$ *performs a polynomially bounded number of queries:*

- Group Setup Query. $\mathcal{A}$ *requests a group secret key* sk_G *for a group* G*. The challenger runs* $sk_G \leftarrow \mathsf{Group\ Setup}(msk, G)$ *and gives* sk_G *to* $\mathcal{A}$ *.*

- Enroll Query. *$\mathcal{A}$ requests a secret key for the member U who belongs to the group G. The challenger runs $sk_{G,U} \leftarrow \mathsf{Enroll}(sk_G, U)$ and gives $sk_{G,U}$ to $\mathcal{A}$.*
- Sign Query. *$\mathcal{A}$ requests a signature on a message M generated by the group G and member identity U. The challenger runs $\sigma \leftarrow \mathsf{Sign}(M, sk_{G,U})$ and returns σ to $\mathcal{A}$.*
- Revocation Query. *$\mathcal{A}$ asks for a revocation token for a member U of the group G. The challenger returns a token $\mathcal{T}_{G,U}$ to $\mathcal{A}$.*

– Challenge. *$\mathcal{A}$ sends to the challenger a message M^*, a group identity G^*, and two member identities U_0 and U_1. $\mathcal{A}$ is restricted in his queries such that $\mathcal{A}$ should not have asked for: a) a group secret key for G^* during* Group Setup Queries*, b) a member secret key for (U_0,U_1) of the group G^* in the* Enroll Query*, and c) a revocation token for (U_0,U_1) of the group G^* in the* Revocation Query*. The challenger picks a random bit $b \in \{0,1\}$, runs $\sigma^* \leftarrow \mathsf{Sign}(M, sk_{G^*}, U_b)$, and returns σ^* to $\mathcal{A}$.*

– Query Phase 2. *$\mathcal{A}$ is allowed to ask additional queries as follows:*

- Group Setup Query. *$\mathcal{A}$ requests a group secret key sk_G for a group G with the restriction that $G \neq G^*$.*
- Enroll Query. *$\mathcal{A}$ requests a secret key for the member U who belongs to group G with the restriction that $G \neq G^* \wedge U \notin \{U_0, U_1\}$.*
- Sign Query. *Same as in* Query Phase 1*.*
- Revocation Query. *Same as in* Query Phase 1 *but $\mathcal{A}$ cannot ask for a revocation token for members U_0 and U_1 of the group G^*.*

– Guess. *$\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ and wins if $b' = b$.*

The advantage of $\mathcal{A}$ in breaking the anonymity property is:

$$\mathsf{ADV}^{anony}_{\mathcal{A},VLR-IBGS}(\lambda) = \left|\Pr[\mathcal{A}\ wins] - \frac{1}{2}\right|$$

where the probability is taken over the random values chosen by $\mathcal{A}$ and the challenger.

The requirement of full traceability captures the notion of unforgeability: the adversary cannot create a valid signature if the group manager cannot trace it to one of the group members. We also formalize this property by considering a security game that involves a challenger and an adversary. First, the challenger runs the setup algorithm and generates a master public key mpk and a master secret key msk, and gives to the adversary the master public key mpk. Next, the adversary can adaptively submit three types of queries: group setup queries, enroll queries, signature queries in the same way as in the anonymous security game. Finally, the adversary is successful if it outputs a tuple $(M^*, \sigma^*, \Re^*, G^*)$, where M^* is a message for which the adversary did not issue a signing query, and σ^* is a valid signature generated by a member of the group G^*, for whom there is no revocation token in $\Re^*$. The following fully-traceable definition is due to Boneh and Shacham [6]:

Definition 3. *The VLR-IBGS scheme is fully-traceable if any $\mathcal{A}$ has only a negligible advantage in the full traceability game which is defined as follows:*

– Setup. *The challenger runs $(mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda)$ and gives mpk to $\mathcal{A}$.*

– Query Phase. *$\mathcal{A}$ performs a polynomially bounded number of* Group Setup Query*,* Enroll Query *and* Sign Query *queries same as in the anonymity game.*

– Forgery Phase. *$\mathcal{A}$ outputs a forgery $(M^*, \sigma^*, \Re^*, G^*)$.*

$\mathcal{A}$ wins the fully-traceability game if: a) $\mathsf{Verify}(M^*, \sigma^*, \Re^*, G^*) = true$*, b) $\mathcal{A}$ did not make a* Sign Query *for (M^*, G^*), c) $\mathcal{A}$ did not make a* Group Setup Query *for G^*, and d) σ^* traces to a member outside $[U] \setminus \Re^*$.*

The advantage of $\mathcal{A}$ in breaking the fully-traceability property:

$$\mathsf{ADV}^{fylly-trace}_{\mathcal{A},VLR-IBGS}(\lambda) = \Pr[\mathcal{A}\ wins]$$

where the probability is taken over the random values chosen by $\mathcal{A}$ and the challenger.

As mentioned by Boneh and Shacham [6], any VLR group signature scheme has an implicit tracing algorithm. The implicit tracing algorithm of our scheme uses the token $\mathcal{T}_{G,U} \in \Re$ to determine which member produced the signature. To determine the identity of the signer producing the signature σ for the message M, the algorithm operates as follows:

- For each member U enrolled in G run: $\mathsf{Verify}(M, \sigma, \Re, G)$.
- Output U of the first member for which $false \leftarrow \mathsf{Verify}\ (M, \sigma, \Re, G)$.

2.2 Complexity Assumptions in Bilinear Groups

Our scheme uses an admissible bilinear map and its security is based on the hardness of the Computational Diffie-Hellman (CDH) and Decisional Linear (DLIN) problems. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative groups of prime order p, and let g be a generator of $\mathbb{G}$. A pairing (or bilinear map) $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ has the following properties [5]:

1. Bilinear: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}^*_p$, we have $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$.
2. Non-degenerate: $\hat{e}(g, g) \neq 1$.

$\mathbb{G}$ is said to be a bilinear group if the group operation in $\mathbb{G}$ and the bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ can be computed efficiently.

Definition 4. *The **Computational Diffie-Hellman Problem** (CDH) in $\mathbb{G}$ is, given elements $(g, g^a, g^b) \in \mathbb{G}$ with $a, b \in \mathbb{Z}_p$, to compute g^{ab}.*

Definition 5. *The **Decisional Linear Problem** (DLIN) in $\mathbb{G}$ is, given a tuple $(g, g_1, g_2, g_1^a, g_2^b, g^c) \in \mathbb{G}$ with $a, b \in \mathbb{Z}_p$, decide whether $c = a + b$ or $c \in_R \mathbb{G}$.*

3 Description of the Scheme

In this section we present the VLR-IBGR scheme that enjoys the security proof in the standard model under the CDH and DLIN assumptions. In a high level, the scheme relies on the presence of a trusted authority (TA) who is in possession of a master key. The TA is responsible for generating system parameters and for creating new groups. A group is managed by a group manager whose responsibility is to enroll new members to the group. The groups are dynamic where new members can join the group after the system parameters are generated. The scheme also allows users to be enrolled in more than one group.

In a high level, our scheme consists from two parallel sub-systems linked via some random variables. The first subsystem is used to show that the signature comes from a member who belongs to a specific group and adapts techniques from Boyen and Waters [7] two-level hierarchical signature scheme in which the first level is the group identity and the second level is the message to be signed, and the second subsystem is used to check whether the group member who creates the signature is revoked or not.

The scheme is based on prime-order bilinear groups. It is important to mention that cryptographic schemes which are based on prime-order bilinear groups are more efficient than

schemes based on composite-order bilinear groups since the size of the prime-order group is smaller than the size of the composite-order group. Due to this fact, group operations on prime-order groups are faster than group operations on composite-order groups.

We build the scheme *VLR-IBGS*=(Setup, Group Setup, Enroll, Sign, Verify) as follows:

- **Setup**(1^λ): TA selects a bilinear group $\mathbb{G}$ of prime order p and elements $g, g_1, g_2 \in_R \mathbb{G}$. It also chooses bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Next to that, the algorithm picks $\alpha, f, t, \chi, \beta, y, y_1, ..., y_k, z, z_1, ..., z_m \in_R \mathbb{Z}_p$.
  The master public key mpk and the master secret key msk are constructed as follows:

$$mpk = \begin{pmatrix} g, g_1, g_2, \hat{e}(g,g)^\alpha, \hat{v}_1 = g_1^t, g^t, \hat{v}_2 = g_2^f, g^f, u = g^y, \\ \{u_i = g^{y_i}\}_{i=1}^k, v = g^z, \{v_j = g^{z_j}\}_{j=1}^m, \hat{e}(g,g)^\chi, g^\beta \end{pmatrix}$$

$$msk = (g^\alpha, g^\chi)$$

  TA stores the master public key mpk in a publicly accessible database and keeps secret the master secret key msk.
  Remarks. The first three components of the mpk are generators of the group $\mathbb{G}$. The fourth component of the msk is used by the verifier to check whether the signature comes from a group member. Part of this component (g^α) is crucial for the system since it is part of the msk and is used by TA to create new groups. The fourth and fifth components (along with g_1 and g_2) are used by the signer to extra randomize the signature and also it helps us to plugin the DLIN assumption and prove the anonymity property of the scheme. The fifth and the six component, are used in the verification phase and are combined with some parts of the signature in order to remove the extra randomness and verify the signature. Same as in [7], we assume that group identities consist of k bits, and messages consist of m bits. Therefore, we have u, u_i, v, v_j components ($k + m + 2$ components) in the mpk which will be used later by the signer and the verifier to represent groups and messages. All these components belong to the first subsystem.
  The last element of the mpk belongs to the second subsystem and is used for revocation purposes in the enrollment phase, signature phase and verification phase.
  The remaining component of the mpk ($\hat{e}(g,g)^\chi$) is used to connect the first subsystem with the second subsystem.
- **Group Setup**(msk, G): To create a secret key for a group represented as a bit string $G = (\kappa_1, ..., \kappa_k) \in \{0,1\}^k$, TA picks ar random $w, \tau \in_R \mathbb{Z}_p$ and outputs a group secret key $sk_G = (\{sk_{G_{(i)}}\}_{1 \le i \le 4})$ where:

$$\begin{aligned} sk_{G_{(1)}} &= g^\alpha \cdot \left(u \prod_{i=1}^{k} u_i^{\kappa_i} \right)^w \\ sk_{G_{(2)}} &= g^w \\ sk_{G_{(3)}} &= g^\chi \cdot g^{\tau w} \\ sk_{G_{(4)}} &= g^\tau \end{aligned}$$

  The TA sends the group secret key sk_G to the group manager through a secure channel.
  Remarks. The group secret key is constructed in such a way that the group manager can change (randomize) w and τ without changing α and χ.

– **Enroll**(sk_G, U): To create a secret key for a member of a group G, the algorithm picks random elements $s, \xi, w', \tau' \in_R \mathbb{Z}_p$, implicitly sets $x = w + w'$, $\tilde{id} = \tau + \tau'$, and outputs a member secret key $sk_{G,U} = (\{sk_{G,U_{(i)}}\}_{1 \leq i \leq 9})$ where:

$$
\begin{aligned}
sk_{G,U_{(1)}} &= sk_{G_{(1)}} \cdot \left(u \prod_{i=1}^{k} u_i^{\kappa_i} \right)^{w'} = g^{\alpha} \cdot \left(u \prod_{i=1}^{k} u_i^{\kappa_i} \right)^{x} \\
sk_{G,U_{(2)}} &= sk_{G_{(2)}} \cdot g^{w'} = g^{x} \\
sk_{G,U_{(3)}} &= sk_{G_{(3)}} \cdot sk_{G_{(4)}}^{w'} \cdot (sk_{G_{(2)}} \cdot g^{w'})^{\tau'} = g^{\chi} \cdot g^{\tilde{id}x} \\
sk_{G,U_{(4)}} &= (\frac{g^{\beta}}{sk_{G_{(4)}} \cdot g^{\tau'}})^{\frac{1}{s}} = g^{\frac{\beta - \tilde{id}}{s}} \\
sk_{G,U_{(5)}} &= (sk_{G_{(2)}} \cdot g^{w'})^{s} = g^{xs} \\
sk_{G,U_{(6)}} &= g^{\xi s} \\
sk_{G,U_{(7)}} &= \left(u \prod_{i=1}^{k} u_i^{\kappa_i} \right)^{\xi} \\
sk_{G,U_{(8)}} &= g^{\xi} \\
sk_{G,U_{(9)}} &= (sk_{G_{(4)}} \cdot g^{\tau'})^{\xi} = g^{\tilde{id}\xi}
\end{aligned}
$$

The group manager sends through a secure channel the member secret key $sk_{G,U}$ to the group member. The group manager keeps a membership table which contains entries (revocation tokens) of the form $\mathcal{T}_{G,U} = (U, g^{\tilde{id}})$. If a group member U is revoked, the group manager publishes the entry of the revoked member. As mentioned above, the entry of the revoked member is stored in the list of revoked members $\Re$ which in turn is stored in a publicly accessible database.

Remarks: We design the secret key in such a way that we allow the signer in the signing phase to directly randomize x without changing (randomizing) α and χ. However, we do not allow the signer to directly randomize $\tilde{id}$ without changing χ. The intuition behind this idea is that we need $\tilde{id}$ in *clear* in order to check whether the verifier is revoked or not. Note that, to achieve unlinkability, the group member randomizes the secret key component which contain $\tilde{id}$ by randomizing other random values. For instance, the signer can randomize $sk_{G,U_{(4)}}$ by randomizing s.

– **Sign**$(M, sk_{G,U})$: To sign a message represented as a binary string $M = (\mu_1, ..., \mu_m) \in \{0,1\}^m$, the signer picks $\rho, s', \xi', \bar{k}, \bar{l} \in_R \mathbb{Z}_p$, implicitly sets $\psi = ss'$ and $\varphi = \xi\xi'$, and outputs the signature $\sigma = (\{\sigma_{(i)}\}_{1 \leq i \leq 8})$ where:

$$
\begin{aligned}
\sigma_{(1)} &= sk_{G,U_{(1)}} \cdot sk_{G,U_{(7)}}^{\xi'} \cdot \left(v \prod_{j=1}^{m} v_j^{\mu_j} \right)^{\rho} \cdot \hat{v}_1^{\bar{k}} \cdot \hat{v}_2^{\bar{l}} \\
&\quad g^{\alpha} \cdot \left(u \prod_{i=1}^{k} u_i^{\kappa_i} \right)^{x+\varphi} \cdot \left(v \prod_{j=1}^{m} v_j^{\mu_j} \right)^{\rho} \cdot \hat{v}_1^{\bar{k}} \cdot \hat{v}_2^{\bar{l}} \\
\sigma_{(2)} &= sk_{G,U_{(2)}} \cdot sk_{G,U_{(8)}}^{\xi'} = g^{x+\varphi} \\
\sigma_{(3)} &= g^{\rho}
\end{aligned}
$$

$$\sigma_{(4)} = g_1^{\bar{k}}$$
$$\sigma_{(5)} = g_2^{\bar{l}}$$
$$\sigma_{(6)} = sk_{G,U_{(3)}} \cdot sk_{G,U_{(9)}}^{\xi'} = g^{\chi} \cdot g^{\tilde{id}(x+\varphi)}$$
$$\sigma_{(7)} = sk_{G,U_{(5)}}^{s'} \cdot sk_{G,U_{(6)}}^{\xi' s'} = g^{(x+\varphi)\psi}$$
$$\sigma_{(8)} = sk_{G,U_{(4)}}^{\frac{1}{s'}} = g^{\frac{\beta - \tilde{id}}{\psi}}$$

– **Verify**$(M, \sigma, \Re, G)$: The verifier performs the following steps in order to check the validity of the signature:

1. **Signature Check.** The verifier checks whether the signer who belongs to a group represented as a bit string $G = (\kappa_1, ..., \kappa_k) \in \{0,1\}^k$ has signed the message represented as a bit string $M = (\mu_1, ..., \mu_m) \in \{0,1\}^m$. Therefore the verifier checks whether the following equation holds:

$$\frac{\hat{e}\left(\sigma_{(1)}, g\right)}{\hat{e}\left(u \prod_{i=1}^{k} u_i^{\kappa_i}, \sigma_{(2)}\right) \cdot \hat{e}\left(v \prod_{j=1}^{m} v_j^{\mu_j}, \sigma_{(3)}\right)} \cdot \frac{1}{\hat{e}\left(\sigma_{(4)}, g^t\right) \cdot \hat{e}\left(\sigma_{(5)}, g^f\right)} \stackrel{?}{=} \hat{e}(g,g)^{\alpha}$$

2. **Revocation Check.** The verifier checks whether the identity of the signer is in the list of revoked members $\Re$ which contains entries (revocation tags) of the form $\mathcal{T}_{G,U} = (U, g^{\tilde{id}})$. First, the verifier computes:

$$a = \frac{\hat{e}(\sigma_{(6)}, g)}{\hat{e}(g,g)^{\chi}}$$

and checks if :

$$a \cdot \hat{e}(\sigma_{(7)}, \sigma_{(8)}) \stackrel{?}{=} \hat{e}(g^{\beta}, \sigma_{(2)})$$

Finally, for each $\mathcal{T}_{G,U} \in \Re$, the verifier checks whether the following equation holds:

$$\hat{e}(g^{\tilde{id}}, \sigma_{(2)}) \stackrel{?}{=} a$$

If the last equation holds then the signer is revoked and the signature is not accepted, i.e. the signature is invalid. Otherwise, the signature is created by a non-revoked user and the signature is accepted, i.e. the signature is valid.

3.1 Correctness

It is easy to proof that the VLR-IDGS satisfies the correctness property. For this reason we have to show that the **Verify** algorithm indeed returns *true* when a signature is created by a non-revoked group member, otherwise the algorithm outputs *false*. If σ is a correctly

generated signature, then the equations under **Signature Check** holds. First we check the equation:

$$\frac{\hat{e}\left(\sigma_{(1)}, g\right)}{\hat{e}\left(u\prod_{i=1}^{k} u_i^{\kappa_i}, \sigma_{(2)}\right)\cdot \hat{e}\left(v\prod_{j=1}^{m} v_j^{\mu_j}, \sigma_{(3)}\right)} \cdot \frac{1}{\hat{e}\left(\sigma_{(4)}, g^t\right)\cdot \hat{e}\left(\sigma_{(5)}, g^f\right)} \stackrel{?}{=} \hat{e}(g,g)^{\alpha} \Rightarrow$$

$$\frac{\hat{e}\left(g^{\alpha}\cdot\left(u\prod_{i=1}^{k} u_i^{\kappa_i}\right)^{x+\varphi}\cdot\left(v\prod_{j=1}^{m} v_j^{\mu_j}\right)^{\rho}\cdot \hat{v}_1^{\bar{k}}\cdot \hat{v}_2^{\bar{l}}, g\right)}{\hat{e}\left(u\prod_{i=1}^{k} u_i^{\kappa_i}, g^{x+\varphi}\right)\cdot \hat{e}\left(v\prod_{j=1}^{m} v_j^{\mu_j}, \sigma_{(3)}\right)} \cdot \frac{1}{\hat{e}\left(g_1^{\bar{k}}, g^t\right)\cdot \hat{e}\left(g_2^{\bar{l}}, g^f\right)} \stackrel{?}{=} \hat{e}(g,g)^{\alpha}$$

Next step is to check the correctness of the **Revocation Check**. We prove that a revoked group member cannot produce a valid group signature. First, we compute:

$$a = \frac{\hat{e}(\sigma_{(6)}, g)}{\hat{e}(g,g)^{\chi}} = \frac{\hat{e}(g^{\chi}\cdot g^{\tilde{id}(x+\varphi)}, g)}{\hat{e}(g,g)^{\chi}} = \hat{e}(g^{\tilde{id}}, g^{x+\varphi})$$

and checks if:

$$a\cdot \hat{e}(\sigma_{(7)}, \sigma_{(8)}) \stackrel{?}{=} \hat{e}(g^{\beta}, \sigma_{(2)}) \Rightarrow$$
$$\hat{e}(g^{\tilde{id}}, g^{x+\varphi})\cdot \hat{e}(g^{\frac{\beta-\tilde{id}}{\psi}}, g^{(x+\varphi)\psi}) \stackrel{?}{=} \hat{e}(g^{\beta}, g^{x+\varphi})$$

Next we prove that if the signature is created by a revoked member, then the last equation under **Revocation Check** holds and the **Verify** outputs *false* i.e. the signature is invalid. Let assume that U_j is a revoked member with a revocation token $(U_j, g^{\tilde{id}_j}) \in \Re$. If U_j creates a signature, then we have:

$$\hat{e}(g^{\tilde{id}_j}, \sigma_{(2)}) \stackrel{?}{=} a$$
$$\hat{e}(g^{\tilde{id}_j}, g^{x+\varphi}) \stackrel{?}{=} \hat{e}(g^{\tilde{id}_j}, g^{x+\varphi})$$

Since this equation holds, the signature is *invalid*.

Finally, we prove that if the signature is created by a non-revoked user then the last equation under **Revocation Check** holds and the **Verify** outputs *true* i.e. the signature is valid. Let assume that $\Re = \{(U_j, g^{\tilde{id}_j})\}$ and the signature is created by a non-revoked user U whose secret key contains the component $\tilde{id}'$. Then we have:

$$\hat{e}(g^{\tilde{id}_j}, \sigma_{(2)}) \stackrel{?}{=} a$$
$$\hat{e}(g^{\tilde{id}_j}, g^{x'+\varphi}) \stackrel{?}{=} \hat{e}(g^{\tilde{id}'}, g^{x'+\varphi})$$

Since this equation does not hold, the signature is accepted as *valid*.

3.2 Efficiency

In terms of efficiency, the size of the signature consists from 8 elements of $\mathbb{G}$ and the creation of a signature requires no pairing operations. An implementation of the scheme using a 256 - bit group order would produce a signature with size of about 256 byte.

3.3 Anonymity Security Proof

In this section prove that the VLR-IDGS has the property of anonymity, assuming that the DLIN problem is hard to be solved.

Theorem 1. *Suppose that there is an algorithm (adversary) $\mathcal{A}$ that wins the anonymity game. Then there is an algorithm $\mathcal{B}$ that solves decision Linear (DLIN) assumption with probability $\hat{\epsilon} = \frac{\epsilon}{2n^2}$.*

Proof. The challenger selects a bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, and generators g, g_1, g_2 of the group $\mathbb{G}$. Then, it picks at random $a, b \in \mathbb{Z}_p$, computes $T_0 = g^{ab}$ and picks at random $T_1 \in_R \mathbb{G}$. It flips a fair coin $\mu \in_R \{0, 1\}$ and gives the DLIN tuple $(g, g_1, g_2, g_1^a, g_2^b, T_\mu) \in \mathbb{G}$ to the reduction $\mathcal{B}$. The goal of $\mathcal{B}$ is to solve the DLIN assumption, and for this reason it runs the algorithm $\mathcal{A}$ as a subroutine in the anonymity game. The reduction $\mathcal{B}$ simulates the game and acts as $\mathcal{A}$'s challenger as follows:
1. Setup. $\mathcal{B}$ picks a fresh uniform $\alpha, \chi, \beta, t, f, y, y_1, ..., y_k, z, z_1, ..., z_m \in_R \mathbb{Z}_p$ and gives to $\mathcal{A}$ the master public key:

$$mpk = \left(\begin{array}{c} g, g_1, g_2, \hat{e}(g,g)^\alpha, \hat{v}_1 = g_1^t, g^t, \hat{v}_2 = g_2^f, g^f, u = g^y, \\ \{u_i = g^{y_i}\}_{i=1}^k, v = g^z, \{v_j = g^{z_j}\}_{j=1}^m, \hat{e}(g,g)^\chi, g^\beta \end{array} \right)$$

The distribution of the mpk in anonymity game is identical to the mpk of the Setup of the scheme since by the DLIN assumption g, g_1, g_2 are random generators of the group $\mathbb{G}$. Further, $\alpha, \chi, \beta, t, f, y, y_1, ..., y_k, z, z_1, ..., z_m$ are chosen at random from $\mathbb{Z}_p$ same as in the actual scheme. Thus mpk as generated by $\mathcal{B}$ has an identical distribution to the output of Setup.
2. Query Phase 1. $\mathcal{A}$ performs a polynomially bounded number of queries:

- Group Setup Query. $\mathcal{A}$ requests a group secret key sk_G for a group G. $\mathcal{B}$ runs $sk_G \leftarrow$ Group Setup(msk, G) same as in the scheme and return sk_G to $\mathcal{A}$.
- Enroll Query. $\mathcal{A}$ requests a secret key for a member U of a group G. For each member $U \notin \{U_0, U_1\}$ of the group G, $\mathcal{B}$ runs $sk_{G,U} \leftarrow$ Enroll(sk_G, U) in the same way as in the scheme and returns $sk_{G,U}$ to $\mathcal{A}$.

  Note that $\mathcal{B}$ does not know the secret keys for members U_0 and U_1. Therefore, if $\mathcal{A}$ requests a secret key for the member U_0 or U_1, $\mathcal{B}$ aborts. However, even if $\mathcal{B}$ aborts, we can define the secret keys for U_0 and U_1 for the rest of the simulation. The secret key for the member

U_0 is defined as:

$$
\begin{aligned}
sk_{G,U_{0(1)}} &= g^{\alpha} \cdot \left(g^{ay} \prod_{i=1}^{k} g^{ay_i\kappa_i} \right) \\
sk_{G,U_{0(2)}} &= g^{a} \\
sk_{G,U_{0(3)}} &= g^{\chi} \cdot g^{a\tilde{id}} \\
sk_{G,U_{0(4)}} &= g^{\frac{\tilde{\beta}-\tilde{id}}{s}} \\
sk_{G,U_{0(5)}} &= g^{as} \\
sk_{G,U_{0(6)}} &= g^{\xi s} \\
sk_{G,U_{0(7)}} &= \left(g^{y} \prod_{i=1}^{k} g^{y_i\kappa_i} \right)^{\xi} \\
sk_{G,U_{0(8)}} &= g^{\xi} \\
sk_{G,U_{0(9)}} &= g^{\tilde{id}\xi}
\end{aligned}
$$

for randomly chosen $\tilde{id}, \xi, s \in_R \mathbb{Z}_p$.
The secret key for the member U_1 is defined as:

$$
\begin{aligned}
sk_{G,U_{1(1)}} &= g^{\alpha} \cdot \left(\frac{T_\mu}{g^b}^{y} \prod_{i=1}^{k} \frac{T_\mu}{g^b}^{y_i\kappa_i} \right) \\
sk_{G,U_{1(2)}} &= \frac{T_\mu}{g^b} \\
sk_{G,U_{1(3)}} &= g^{\chi} \cdot \frac{T_\mu}{g^b}^{\tilde{id}} \\
sk_{G,U_{1(4)}} &= g^{\frac{\tilde{\beta}-\tilde{id}}{s}} \\
sk_{G,U_{1(5)}} &= \left(\frac{T_\mu}{g^b} \right)^{s} \\
sk_{G,U_{1(6)}} &= g^{\xi s} \\
sk_{G,U_{1(7)}} &= \left(g^{y} \prod_{i=1}^{k} g^{y_i\kappa_i} \right)^{\xi} \\
sk_{G,U_{1(8)}} &= g^{\xi} \\
sk_{G,U_{1(9)}} &= g^{\tilde{id}\xi}
\end{aligned}
$$

for randomly chosen $\tilde{id}, \xi, s \in_R \mathbb{Z}_p$.
If $\mu = 0$ and $T_0 = g^{a+b}$ then for both users, the components $sk_{G,U_{0/1(1)}}$ and $sk_{G,U_{0/1(2)}}$ are same. But, if μ_1 and $T_1 \in_R \mathbb{Z}_p$, then both users have independent private keys in the same way as explained in the anonymity game.

– Sign Query. $\mathcal{A}$ requests a signature on a message $M = (\mu_1, ..., \mu_m) \in \{0,1\}^m$ generated by U who is a member of the group $G = (\kappa_1, ..., \kappa_k) \in \{0,1\}^k$. The algorithm $\mathcal{B}$ may operate in the following ways:

1. If $U \notin \{U_0, U_1\}$, $\mathcal{B}$ runs $\sigma \leftarrow \mathsf{Sign}(M, sk_{G,U})$ in the same way as in the scheme and returns σ to $\mathcal{A}$.
2. If $U = U_0$, $\mathcal{B}$ picks a fresh uniform $\tilde{id}, a', \psi, \rho, \bar{k}, \bar{l} \in_R \mathbb{Z}_p$, computes $\frac{g_1^{a'}}{g_1^a} = g_1^{\varphi}$ (thus $a' = a + \varphi$) and computes the signature as follows:

$$\begin{aligned}
\sigma_{(1)} &= g^{\alpha} \cdot \left(g^{(a+\varphi)y} \prod_{i=1}^{k} g^{(a+\varphi)y_i\kappa_i} \right) \cdot \left(v \prod_{j=1}^{m} v_j^{\mu_j} \right)^{\rho} \\
&\quad \cdot g_1^{at\bar{k}} \cdot g_2^{bf\bar{l}} \\
\sigma_{(2)} &= g^{a'} = g^{a+\varphi} \\
\sigma_{(3)} &= g^{\rho} \\
\sigma_{(4)} &= g_1^{a\bar{k}} \\
\sigma_{(5)} &= g_2^{b\bar{l}} \\
\sigma_{(6)} &= g^{\chi} \cdot g^{a'\tilde{id}} = g^{\chi} \cdot g^{(a+\varphi)\tilde{id}} \\
\sigma_{(7)} &= g^{a'\psi} = g^{(a+\varphi)\psi} \\
\sigma_{(8)} &= g^{\frac{\beta - \tilde{id}}{\psi}}
\end{aligned}$$

   The simulated signature has the same distribution as the signature generated in the real scheme. Note that $\tilde{id}, \psi, \rho, \bar{k}, \bar{l} \in_R \mathbb{Z}_p$ are random values and their distribution is same as in the real scheme. In the simulated signature we have set x to be equal to a and since the latter is chosen at random and is taken from the DLIN assumption, then we conclude that the distribution of components which contain a in the simulation is same as the distribution of components which contain x in the scheme.
3. If $U = U_1$, $\mathcal{B}$ picks a fresh uniform $\tilde{id}, b', \psi, \rho, \bar{k}, \bar{l} \in_R \mathbb{Z}_p$, computes $g_2^{b'} g_2^{b} = g_2^{\varphi}$ (thus $b' = \varphi - b$) and computes the signature as follows:

$$\begin{aligned}
\sigma_{(1)} &= g^{\alpha} \cdot \left(\frac{T_\mu}{g^b}^{y} \prod_{i=1}^{k} \frac{T_\mu}{g^b}^{y_i\kappa_i} \right) \cdot g^{\varphi y} \cdot \prod_{i=1}^{k} g^{\varphi y_i \kappa_i} \\
&\quad \cdot \left(v \prod_{j=1}^{m} v_j^{\mu_j} \right)^{\rho} \cdot g_1^{at\bar{k}} \cdot g_2^{bf\bar{l}} \\
\sigma_{(2)} &= T_\mu \cdot g^{b'} = \frac{T_\mu}{g^b} \cdot g^{\varphi} \\
\sigma_{(3)} &= g^{\rho} \\
\sigma_{(4)} &= g_1^{a\bar{k}} \\
\sigma_{(5)} &= g_2^{b\bar{l}} \\
\sigma_{(6)} &= g^{\chi} \cdot T_\mu^{\tilde{id}} \cdot g^{\tilde{id}b'} = g^{\chi} \cdot \frac{T_\mu}{g^b}^{\tilde{id}} \cdot g^{\tilde{id}\varphi} \\
\sigma_{(7)} &= (T_\mu \cdot g^{b'})^{\psi} = (\frac{T_\mu}{g^b} \cdot g^{\varphi})^{\psi} \\
\sigma_{(8)} &= g^{\frac{\beta - \tilde{id}}{\psi}}
\end{aligned}$$

The simulated signature has the same distribution as the signature generated in the real scheme. Note that $\tilde{id}, \psi, \rho, \bar{k}, \bar{l} \in_R \mathbb{Z}_p$ are random values and their distribution is same as in the real scheme. In the simulated signature we have set x to be equal to $c-b$ (assuming that $T_\mu = g^c$ and the exponent c can be either equal to ab or a random value from $\mathbb{Z}_p$) and since the discrete log of T_μ to the base g is chosen at random and is taken from the DLIN assumption, then we conclude that the distribution of components which contain $c-b$ in the simulation is same as the distribution of components which contain x in the scheme.

- Revocation Query. $\mathcal{A}$ asks for a revocation token for a member U_i of a group G. $\mathcal{B}$ aborts if $\mathcal{A}$ asks for a revocation token for the member U_0 or U_1. $\mathcal{B}$ returns a token $\mathcal{T}_{G,U_i}$ to $\mathcal{A}$.

3. Challenge. $\mathcal{A}$ returns to $\mathcal{B}$ two tuples: (M, G^*, U_{0^*}) and (M, G^*, U_{1^*}). If $U_{0^*} \neq U_0$ and $U_{1^*} \neq U_1$, then $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ picks a random bit $b \in \{0,1\}$ and runs $\sigma^* \leftarrow \mathsf{Sign}(M, sk_{G^*}, U_{b^*})$ in the same way as explained under Sign Query. $\mathcal{B}$ returns σ^* to $\mathcal{A}$.
4. Query Phase 2. The adversary $\mathcal{A}$ issues *restricted* queries, as defined in Definition 2, and the reduction $\mathcal{B}$ replies as in Query Phase 1.
5. Guess. $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$, and if $b' = b$ then $\mathcal{B}$ outputs 0 and $T_\mu = g^{a+b}$, otherwise $\mathcal{B}$ outputs 1 and T_μ is a random element chosen from $\mathbb{G}$.

Suppose $\mathcal{B}$ does not abort during the simulation. When $\mu = 0$ and $T_0 = g^{a+b}$, then secret keys components $sk_{G,U_{(1)}}$ and $sk_{G,U_{(2)}}$ for users U_0 and U_1 are same and the challenge signature is independent of b. Thus $\Pr[b' = b] = \frac{1}{2}$. When $\mu = 1$ and $T \in_R \mathbb{G}$, then secret keys for users U_0 and U_1 are independent and generated in the same was as in the anonymity game. Thus $\Pr[b' = b] > \frac{1}{2} + \epsilon$.

Assuming that $\mathcal{B}$ does not abort in the simulation, the overall advantage to solve DLIN assumption is $\frac{\epsilon}{2}$. $\mathcal{B}$ does not abort if correctly guesses the identities U_0 and U_1 and none of the queries in the Query Phase 1 and the choice of the challenge does not cause $\mathcal{B}$ to abort. The probability that queries in the Query Phase 1 and the choice of challenge does not cause $\mathcal{B}$ to abort is at least $\frac{1}{n^2}$, where n is the number of members in the scheme. Therefore, we conclude that $\mathcal{B}$ solves DLIN problem with advantage at least $\frac{\epsilon}{2n^2}$. □

3.4 Full traceability Security Proof

In this section we prove the property of traceability assuming that the CDH problem is hard to be solved. To prove this property we closely follow the security analysis from [7].

Theorem 2. *Suppose that there is an algorithm (adversary) $\mathcal{A}$, in an adaptive chosen message attack, that after l signature queries in the full traceability game creates a forgery with a non-negligible advantage ϵ . Then there is an algorithm $\mathcal{B}$ that solves CDH assumption with probability $\hat{\epsilon} \geq \frac{\epsilon}{2^{k+2}ml}$.*

Proof. The challenger selects a bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, and the generator g of the group $\mathbb{G}$. Then, it picks at random $a, b \in \mathbb{Z}_p$ and hands the CDH instance (g, g^a, g^b) to the reduction $\mathcal{B}$. The reduction $\mathcal{B}$ receives the CDH instance and solves the CDH problem (computes g^{ab}) by running the algorithm $\mathcal{A}$ as a subroutine. The reduction $\mathcal{B}$ acts as $\mathcal{A}$'s challenger in the full traceability game and we show that if $\mathcal{A}$ produces a forgery, then $\mathcal{B}$ can use that forgery to solve the CDH problem. The game proceeds as follows:
1. Setup. Let $G^* = \{\kappa_1^*, ..., \kappa_k^*\} \in \{0,1\}^k$ be the group for which $\mathcal{A}$ wants to create a forgery. The algorithm $\mathcal{B}$ chooses a random number $k \in \{0, ..., m\}$ and random numbers $x, x_1 ..., x_m$

from the interval $\{0, ..., 2l-1\}$. Next to that, $\mathcal{B}$ chooses at random the generator $g_1 \in \mathbb{G}$ and exponents $\beta, \chi, \hat{y}, \hat{y}_1, ..., \hat{y}_k, \hat{z}, \hat{z}_1, ..., \hat{z}_m, t, f, W \in_R \mathbb{Z}_p$, it sets $y + \sum_{i=1}^{k} \kappa_i^* y_i \equiv 0 \ (mod\ p)$ and $g_2 = g^b$, and outputs the master public key:

$$mpk = \begin{pmatrix} g, g_1, g_2 = g^b, \hat{e}(g^a, g^b), \hat{v}_1 = g_1^t, g^t, \hat{v}_2 = g_2^f, \\ g^f, u = g^{a\hat{y}} g^W, \{u_i = g^{a\hat{y}_i}\}_{i=1}^k, v = g_2^{x-2kl} g^{\hat{z}}, \\ \{v_j = g_2^{x_j} g^{\hat{z}_j}\}_{j=1}^m, \hat{e}(g,g)^\chi, g^\beta \end{pmatrix}$$

The mpk generated by $\mathcal{B}$ has the same distribution as the mpk generated by Setup of the scheme. Note that since b is chosen at random from $\mathbb{Z}_p$ (b comes from the CDH assumption) then g, g_1, g_2 are random generators of the group $\mathbb{G}$ in the view of $\mathcal{A}$. If we set $\alpha = ab$, $y = a\hat{y} + W$,$y_i = a\hat{y}_i$, $z = b(x-2kl) + \hat{z}$ and $z_j = bx_j + \hat{z}_j$ then values α, y, y_i, z, z_j have the same distribution as in the scheme since $a, b, \hat{y}, \hat{y}_i, \hat{z}, \hat{z}_j, x, x_1 ... x_m, W$ are chosen uniformly at random from $\mathbb{Z}_p$. Finally, the values β, χ, t, f are chosen in the same way as in the scheme.
2. Query Phase. $\mathcal{A}$ performs a polynomially bounded number of queries:

- Group Setup Query. $\mathcal{A}$ requests a group secret key sk_G for a group $G = \{\kappa_1, ..., \kappa_k\}$. Let $T = \hat{y} + \sum_{i=1}^{k} \kappa_i \hat{y}_i$. The challenger $\mathcal{B}$ picks $\tau, \bar{z} \in_R \mathbb{Z}_p$ and computes $\frac{g^{\bar{z}}}{(g^b)^{\frac{1}{T}}} = g^w$ (thus $\bar{z} = w + \frac{b}{T}$). $\mathcal{B}$ returns to $\mathcal{A}$ the group secret key:

  $$sk_{G_{(1)}} = g^{\frac{-Wb}{T}} \left(g^W g^{aT}\right)^{\bar{z}} \qquad sk_{G_{(2)}} = g^w$$
  $$sk_{G_{(3)}} = g^\chi \cdot g^{\tau w} \qquad sk_{G_{(5)}} = g^\tau$$

  The group secret key sk_G generated by $\mathcal{B}$ in the security game and the sk_G generated by **Group Setup** of scheme have the same distribution since $w = \bar{z} - \frac{b}{T}$ is a random value ($\bar{z}$ is chosen at random) in the view of $\mathcal{A}$, and $\tau \in_R \mathbb{Z}_p$ is chosen in the same way as in the scheme.
- Enroll Query. $\mathcal{A}$ requests a secret key for a member U of a group G. $\mathcal{B}$ runs $sk_{G,U} \leftarrow \mathsf{Enroll}(sk_G, U)$ same as in the scheme and returns $sk_{G,U}$ to $\mathcal{A}$. The group secret key sk_G is computed in the same way as explained under Group Setup Query.
- Sign Query. $\mathcal{A}$ requests a signature on a message $M = (\mu_1, ..., \mu_m)$ generated by a member U of a group G. $\mathcal{B}$ may operate in the following two ways:
  I. If $G \neq G^*$, $\mathcal{B}$ runs $\sigma \leftarrow \mathsf{Sign}(M, sk_{G,U})$ in the same way as in the scheme.
  II. If $G = G^*$, then let $F = -2kl + x + \sum_{j=1}^{m} x_j \mu_j$ and $J = \hat{z} + \sum_{j=1}^{m} \hat{z}_j \mu_j$. If $F = 0$, then $\mathcal{B}$ aborts because it cannot simulate the signature. Otherwise, $\mathcal{B}$ chooses $x, \psi, \varphi, q, \tilde{id}, \bar{k}, \bar{l} \in_R \mathbb{Z}_p$ and sets $\frac{g^q}{(g^a)^{\frac{1}{F}}} = g^\rho$ (thus $q = \rho + \frac{a}{F}$). $\mathcal{B}$ returns to $\mathcal{A}$ the signature on M:

$$\begin{aligned}
\sigma_{(1)} &= g^{\frac{-aJ}{F}} \cdot \left(g^W\right)^{x+\varphi} \cdot \left(g^J g_2^F\right)^q \cdot \hat{v}_1^{\bar{k}} \cdot \hat{v}_2^{\bar{l}} \\
&= g^{\frac{-aJ}{F}} \cdot \left(g^W\right)^{x+\varphi} \cdot \left(g^J g_2^F\right)^{\rho + \frac{a}{F}} \cdot \hat{v}_1^{\bar{k}} \cdot \hat{v}_2^{\bar{l}} \\
&= g^{\frac{-aJ}{F}} \cdot (g^W)^{x+\varphi} \cdot \left(g^J g_2^F\right)^\rho \cdot \left(g^J g_2^F\right)^{\frac{a}{F}} \cdot \hat{v}_1^{\bar{k}} \cdot \hat{v}_2^{\bar{l}} \\
&= g^{ab} \cdot \left(g^W\right)^{x+\varphi} \cdot \left(g^J g_2^F\right)^\rho \cdot \hat{v}_1^{\bar{k}} \cdot \hat{v}_2^{\bar{l}} \\
\sigma_{(2)} &= g^{x+\varphi} \\
\sigma_{(3)} &= g^\rho \\
\sigma_{(4)} &= g_1^{\bar{k}} \\
\sigma_{(5)} &= g_2^{\bar{l}}
\end{aligned}$$

$$\sigma_{(6)} = g^{\chi} \cdot g^{\tilde{id}(x+\varphi)}$$
$$\sigma_{(7)} = g^{(x+\varphi)\psi}$$
$$\sigma_{(8)} = g^{\frac{\beta - \tilde{id}}{\psi}}$$

The signature σ generated by $\mathcal{B}$ in the security game has the same distribution as the signature generated by **Sign** of the scheme. Note that ρ depends on q which is uniformly at random chosen from $\mathbb{Z}_p$, therefore the entire value of ρ is random in the view of $\mathcal{A}$, same as in the scheme. Finally, the values $x, \psi, \varphi, \tilde{id}, \bar{k}, \bar{l}$ are chosen uniformly random same as in the scheme.

3. Forgery. $\mathcal{A}$ outputs a valid forgery $(M^*, \sigma^*, \Re^*, G^*)$ where $F^* = 0 \ (mod\ p)$ and $J = \hat{z} + \sum_{j=1}^{m} \hat{z}_j \mu_j^*$. If $F^* \neq 0 (mod\ p)$ then $\mathcal{B}$ aborts. Note that a valid signature σ^* has the following form:

$$\sigma^*_{(1)} = g^{ab} \cdot (g^W)^{x+\varphi} \cdot g^{J\rho} \cdot \hat{v}_1^{\bar{k}} \cdot \hat{v}_2^{\bar{l}}$$
$$\sigma^*_{(2)} = g^{x+\varphi}$$
$$\sigma^*_{(3)} = g^{\rho}$$
$$\sigma^*_{(4)} = g_1^{\bar{k}}$$
$$\sigma^*_{(5)} = g_2^{\bar{l}}$$
$$\sigma^*_{(6)} = g^{\chi} \cdot g^{\tilde{id}(x+\varphi)}$$
$$\sigma^*_{(7)} = g^{(x+\varphi)\psi}$$
$$\sigma^*_{(8)} = g^{\frac{\beta - \tilde{id}}{\psi}}$$

$\mathcal{B}$ solves the CDH problem as follows:

$$\sigma^*_{(1)} \cdot \sigma^{*-W}_{(2)} \cdot \sigma^{*-J}_{(3)} \cdot \sigma^{*-t}_{(4)} \cdot \sigma^{*-f}_{(5)}$$
$$= g^{ab} \cdot (g^W)^{x+\varphi} \cdot g^{J\rho} \cdot \hat{v}_1^{\bar{k}} \cdot \hat{v}_2^{\bar{l}} \cdot (g^{x+\varphi})^{-W} \cdot (g^{\rho})^{-J}$$
$$\cdot \hat{v}_1^{-\bar{k}} \cdot \hat{v}_2^{-\bar{l}} = g^{ab}$$

$\mathcal{B}$ does not abort if in the Setup phase correctly guesses the group $G^* = \{\kappa_1^*, ..., \kappa_k^*\}$, in the Sign Query the $F \neq 0 (mod\ p)$, and in the Forgery phase the $F^* \equiv 0 (mod\ p)$. The probability that $\mathcal{B}$ in the Setup phase guesses G^* is $\frac{1}{2^k}$. The probability that for each individual Sign Query the $F^* \neq 0 (mod\ p)$ is $1 - \frac{1}{2l}$, therefore the total probability for l queries is larger than $\frac{1}{2}$, and the probability that $F^* \equiv 0 (mod\ p)$ is $\frac{1}{2ml}$. Since, the advantage of $\mathcal{A}$ is ϵ, $\mathcal{B}$ solves CDH assumption with probability $\hat{\epsilon} \geq \frac{\epsilon}{2^{k+2} ml}$.□

4 Conclusion

We propose a verifier-local revocation identity-based group signature (VLR-IBGS) scheme based on prime order bilinear groups with a security proof under standard assumptions. Indeed, this is the first VLR group signature scheme which achieves simultaneously three desirable properties: supporting membership revocation, having a security proof in the standard model and being identity-based group signature scheme where the group public key is derived

from the group identity. We prove that the scheme has the property of anonymity under the Decisional Linear (DLIN) assumption and that it is fully-traceable under the Computational Diffie-Hellman (CDH) assumption.

References

1. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In M. Bellare, editor, *Proceedings of Crypto 2000*, volume 1880 of *LNCS*, pages 255–270. Springer, 2000.
2. G. Ateniese, D. Song, and G. Tsudik. Quasi-efficient revocation of group signatures. In M. Blaze, editor, *Proceedings of Financial Cryptography 2002*, volume 2357 of *LNCS*, pages 183–197. Springer, 2003.
3. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In E. Biham, editor, *Proceedings of Eurocrypt 2003*, volume 2656 of *LNCS*, pages 614–629. Springer, 2003.
4. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *Proceedings of Crypto 2002*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.
5. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In J. Kilian, editor, *Proceedings of Crypto 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.
6. D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In B. Pfitzmann and P. Liu, editors, *Proceedings of CCS 2004*, pages 168–177. ACM, 2004.
7. X. Boyen and B. Waters. Compact group signatures without random oracles. In S. Vaudenay, editor, *Proceedings of Eurocrypt 2006*, volume 4004 of *LNCS*, pages 427–444. Springer, 2006.
8. X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In T. Okamoto and X. Wang, editors, *Proceedings of PKC 2007*, volume 4450 of *LNCS*, pages 1–15. Springer, 2007.
9. C. Brzuska, M. Fischlin, A. Lehmann, and D. Schroder. Unlinkability of Sanitizable Signatures. In P.Q. Nguyen and D Pointcheval, editors, *Proceedings of PKC 2010*, volume 6056 of *LNCS*, pages 444–461. Springer, 2010.
10. J. Camenisch. Efficient and generalized group signatures. In V. Fumy, editor, *Proceedings of Eurocryp 1997*, volume 1233 of *LNCS*, pages 465–479. Springer, 1997.
11. J. Camenisch and J. Groth. Group signatures: Better efficiency and new theoretical aspects. In C. Blundo and S. Cimato, editors, *Proceedings of Security in Communication Networks*, volume 3352 of *LNCS*, pages 120–133. Springer, 2004.
12. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In M. Yung, editor, *Proceedings of Crypto 2002*, volume 2442 of *LNCS*, pages 61–76. Springer, 2002.
13. R. Canetti, O. Goldreich, and S. Halevi. The Random Oracle Methodology, Revisited. *Journal of the ACM (JACM)*, 51(4):557–594, 2004.
14. D. Chaum and E. Van Heyst. Group signatures. In D. W. Davies, editor, *Proceedings of Eurocrypt 1991*, volume 547 of *LNCS*, pages 257–265. Springer, 1991.
15. C. Gentry and A. Silverberg. Hierarchical id-based cryptography. In Y. Zheng, editor, *Proceedings of Asiacrypt 2002*, volume 2501 of *LNCS*, pages 548–566. Springer, 2002.
16. J. Groth. Fully anonymous group signatures without random oracles. In K. Kurosawa, editor, *Proceedings of Asiacrypt 2007*, volume 4833 of *LNCS*, pages 164–180. Springer, 2007.
17. P. Gutman. Pki: It's not dead, just resting. In *IEEE Computer*, volume 35 of *IEEE Computer*, pages 41–49, 2002.
18. A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In R. Cramer, editor, *Proceedings of Eurocrypt 2005*, volume 3494 of *LNCS*, pages 198–214. Springer, 2005.
19. B. Libert and D. Vergnaud. Group signatures with verifier-local revocation and backward unlinkability in the standard model. In J. Garay, A. Miyaji, and A. Otsuka, editors, *Cryptology and Network Security 2009*, volume 5888 of *LNCS*, pages 498–517. Springer, 2009.
20. T. Nakanishi and N. Funabiki. Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. In R. Bimal, editor, *Proceedings of Asiacrypt 2005*, volume 3788 of *LNCS*, pages 533–548. Springer, 2005.
21. A. Shamir. Identity-based cryptosystems and signature schemes. In G.R. Blakely and D. Chaum, editors, *Proceedings of Crypto 1984*, volume 196 of *LNCS*, pages 47–53. Springer, 1985.

22. N.P. Smart and B. Warinschi. Identity based group signatures from hierarchical identity-based encryption. In H Shacham and B Waters, editors, *Proceedings of Pairing 2009*, volume 5671 of *LNCS*, pages 150–170. Springer, 2009.
23. D.X. Song. Practical forward secure group signature schemes. In M Reiter and P. Samarati, editors, *Proceedings of CCS 2001*, pages 225–234. ACM, 2001.