# Hierarchical Group Signatures with Verifier-Local Revocation

Lin Hou[1,2(✉)], Renzhang Liu[3], Tian Qiu[1,2], and Dongdai Lin[1,2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{houlin,qiutian,ddlin}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] Westone Cryptologic Research Center, Westone Information Industry INC., Beijing, China
liu.renzhang05391@westone.com.cn

**Abstract.** Group signatures are important when it comes to authentication with privacy. Hierarchical group signatures, as a proper generalization of group signatures, have splendid applications in e-commerce. One key issue for such schemes is to support membership revocation in an efficient as well as secure way. To this end, the notion of group signatures with verifier-local revocation was proposed and well-studied, where the revocation messages are sent only to verifiers. However, such issue has not been formally studied in the context of hierarchical group signatures. In this paper, we raise and formalize the new notion of hierarchical group signatures with verifier-local revocation, and propose a semi-generic construction from group signatures with verifier-local revocation. When instantiating it with a variant of the group signature scheme proposed by Gordon, Katz and Vaikuntanathan, a lattice-based construction is implicitly given.

**Keywords:** Group signatures · Authentication with privacy
Lattice-based cryptography

## 1 Introduction

Digital signatures are ubiquitous as a main approach for authentication. However, ordinary digital signatures (via PKI) inherently expose signers' identities, and such privacy is much desired in many real-world scenarios, e.g., e-commerce, e-cash, anonymous online communications and more. To solve this issue, several privacy-oriented signatures were proposed, such as ring signatures [25], traceable signatures [13], domain-specific pseudonymous signatures [6], and especially, *group signatures* (GS) [9]. Loosely speaking, a group signature scheme has both anonymity and traceability. The former means that group members can sign on behalf of the group, without leaking out their identities; on the other hand, given some valid message-signature pair, traceability enables some designated


D. Naccache et al. (Eds.): ICICS 2018, LNCS 11149, pp. 271–286, 2018.
https://doi.org/10.1007/978-3-030-01950-1_16

manager to run a open algorithm using some secret tracing key and figure out the actual signer.

Up to now, many generalized notions of group signatures were proposed, such as sub-group signatures [4], multi-group signatures [4], group blind signatures [22] and *hierarchical group signatures* (HGS) [26]. Hierarchical group signatures was brought up by Wikström *et al.* in the context of anonymous credit card systems. Imagine a balanced tree of depth two. The root stands for some payment network, and nodes at depth one are distinct card-issuing banks, while leaves are their users (card-holders). In a transaction, a user signs on the transaction information to generate a signature, of which the validity with respect to some *single* public verification key can be easily checked by the merchant; the merchant sends the message-signature pair to the payment network, and the latter will figure out then route it to the user's bank; eventually, the bank traces to the user, and debits its account. One admirable feature of such system is that by such hierarchical tracing, nothing except those absolutely necessary will be revealed to each party: the merchant is convinced that the transaction information has been signed by some valid user, but cannot know its issuing bank (let alone its identity); the payment network can route the transaction to the user's bank, but infeasible to figure out its identity; in contrast, the bank must be able to determine the exact identity to debit the correct account.

**Related Work.** In their foundational work [5], Bellare *et al.* formalized two properties for static group signatures, namely *full-anonymity* and *full-traceability*. Plenty of subsequent work has been done within this framework. They also proposed a generic GS construction from trapdoor permutations, which essentially reflects a *sign-encrypt-proof* designing paradigm.

On the other hand, lattice-based cryptography [1] has seen a flourish of research works in recent years. In our interest, Gordon, Katz and Vaikuntanathan gave the first group signature scheme from lattice assumptions [12] (abbreviated as the GKV scheme), and we refer to the full version or the original paper for a detailed description of their scheme; besides, there are several lattice-based schemes [8,14–21,23] with different security models, different levels of efficiency and functionality.

Wikström *et al.* introduced the notion of hierarchical group signatures [26]. Without loss of generality, they considered *a balanced tree* depicting the hierarchy, where inner nodes are managers and leaves are signers. Given a valid message-signature pair, *a path-following tracing*, namely an iterative process where some father node (initialized as the root manager) traces then routes the pair to some child node, will always locate some signer who is (amongst) the actual generator(s); on the other hand, nobody can non-trivially figure out the actual signer without such hierarchical tracing. These are formalized into the *traceability* and *anonymity* properties for HGS in the framework of Bellare *et al.* [5]. Moreover, Wikström *et al.* gave a generic construction assuming the existence of a family of trapdoor permutations.

**Motivations.** Wikström *et al.* did foundational works for HGS in the static setting, where no dynamic joining or revocation will be allowed once the system is set up. However in practice, there are scenarios where revocation is desired or even necessary, for example, when some signer misbehaves or accidently exposes its secret signing key and thus has to be removed from the original hierarchy. *Verifier-local revocation* is a highly efficient approach early brought up in the setting of group signatures, with which revocation messages are *only* sent to signature verifiers. Naturally, we wonder how to make an HGS scheme *efficiently revocable*.

**Our Contributions and Main Techniques.** We formalize the new notion of *hierarchical group signatures with verifier-local revocation* (HGS-VLR) and propose a semi-generic construction from the existing notion of *group signatures with verifier-local revocation* (GS-VLR). Moreover, we implicitly provide an instantiation from lattice assumptions, using a variant of the GKV group signature scheme.

An HGS-VLR scheme consists of three algorithms: a key generation algorithm, a signing algorithm and a verification algorithm. Unlike regular HGS, HGS-VLR has no explicit tracing algorithm. A father node possessing all children's revocation tokens can trace implicitly using the verification algorithm. Correctness says that for any honestly generated message-signature pair, it will pass the verification if and only if no ancestor of the signer (including itself) has been revoked. As for anonymity, we propose the full-version of anonymity for HGS-VLR where the adversary is given all signers' secret keys. In contrast, an insider-version of anonymity is usually considered in the context of GS-VLR, namely the adversary cannot obtain the secret keys of challenge identities.

The main difficulties lie in how to properly define the traceability property for HGS-VLR. If we stick to the original path-following tracing as in defining traceability for HGS, inconsistencies will occur. Instead, we introduce *a whole-depth tracing* in the model, which involves *all managers at the penultimate depth* in a joint and independent tracing of some valid message-signature pair. In our model, the adversary is claimed a success if it comes up with some valid message-signature pair, and it holds either all managers at the penultimate depth cannot open this pair, or there exists some manager at the penultimate depth traces it to some honest signer. The implications and rationalities of the traceability property as such defined will be detailed in Sect. 4.

With our new notions and models, a semi-generic HGS-VLR construction from GS-VLR naturally arises. We regard all parties at the same depth as a group respectively, and generate keys and revocation tokens for them. A signer is given all signing keys of its ancestors (including itself), and for tracing purposes, a manager is given all revocation tokens of its direct children. To generate a signature, the signer produces compositional group signatures for all its ancestors. The correctness, anonymity and traceability can be easily reduced to those of the underlying GS-VLR respectively.

**Outline of This Paper.** We first recall some lattice knowledge. In Sect. 3, we recall the notion of GS-VLR, and show that a variant of the GKV scheme is a fully-anonymous GS-VLR scheme. In Sect. 4, we introduce and formalize our new notion, and give a semi-generic construction. Its GKV instantiation is straightforward, when combining work done in Sect. 3. We conclude this paper in Sect. 5. Due to lack of spaces, we refer interested readers to the full version for all proofs.

## 2 Preliminaries

In this section, we briefly introduce some background on lattice.

### 2.1 Notations

Let $x\|y$ denote the concatenation of two binary strings $x$ and $y$. Vectors are assumed to be in column form and are written using bold lower-case letters, e.g. $\mathbf{x}$, and let $\|\mathbf{x}\|$ denote the Euclidean norm of a vector $\mathbf{x}$. Matrices are written as bold capital letters. For a matrix $\mathbf{X}$, let $\|\mathbf{X}\|$ denote the maximum of the Euclidean norms of the columns of $\mathbf{X}$ and $\widetilde{\mathbf{X}}$ is the *Gram-Schmidt orthogonalization* of $\mathbf{X}$.

We denote the set $\{1, \cdots, N\}$ by $[N]$, where $N \in \mathbb{N}$. If $S$ is some finite set, we denote its cardinality by $|S|$, and denote choosing a uniformly random element from $S$ by $s \xleftarrow{\$} S$; the uniform distribution on $S$ is denoted $\mathsf{U}(S)$. If $A$ is a randomized algorithm, then $[A(x, y, \dots)]$ denotes the set of all outputs having positive probability on inputs $x, y, \dots$. We use the standard big$-O$ notation to classify the growth of functions. **Oracles** are written bold to be distinguished from algorithms.

### 2.2 Lattices

**Definition 1 (Lattice).** *Let $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ be $n$ ($\leqslant m$) linearly independent vectors in $\mathbb{R}^m$. The **lattice** generated by $\mathbf{B}$, denoted by $\mathcal{L}(\mathbf{B})$, is the set of all the integer linear combination of the vectors in $\mathbf{B}$, and $\mathbf{B} \in \mathbb{R}^{m \times n}$ is called **a basis** of $\mathcal{L}(\mathbf{B})$. Namely, $\mathcal{L}(\mathbf{B}) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i \mid x_i \in \mathbb{Z}\} = \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^n\}$.*

**Definition 2 (Shortest Vector Problem, SVP).** *Given a basis $\mathbf{B} \in \mathbb{R}^{m \times n}$ of $\mathcal{L}(\mathbf{B})$, find the shortest nonzero vector in $\mathcal{L}(\mathbf{B})$, denoted by $\lambda_1(\mathcal{L}(\mathbf{B}))$.*

It has been proved that the SVP problem is NP-hard under randomized reduction [2]. We use its promise variant, namely the $\mathsf{GapSVP}_\gamma$ problem.

**Definition 3 ($\mathsf{GapSVP}_\gamma$).** *An instance of the problem is given by a pair $(\mathbf{B}, r)$ where $\mathbf{B} \in \mathbb{Z}^{m \times n}$ is a lattice basis and $r \in \mathbb{Q}$. In YES instance, $\lambda_1(\mathcal{L}(\mathbf{B})) \leqslant r$. In NO instance, $\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma \cdot r$. The goal is to determine which case the input instance is.*

Two classes of random lattices are widely used in cryptography:

**Definition 4** ($\mathcal{L}^{\perp}(\mathbf{B})$)**.** *Fixing $q, m, n \in \mathbb{N}$ and given a matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$, the $m$-dimensional lattice $\mathcal{L}^{\perp}(\mathbf{B})$ is defined as:*

$$\mathcal{L}^{\perp}(\mathbf{B}) = \{\omega \in \mathbb{Z}^m \mid \mathbf{B}\omega \equiv 0 \pmod q\}.$$

**Definition 5** ($\mathcal{L}(\mathbf{B}^T)$)**.** *Fixing $q, m, n \in \mathbb{N}$ and given a matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$, the $m$-dimensional lattice $\mathcal{L}(\mathbf{B}^T)$ is defined as:*

$$\mathcal{L}(\mathbf{B}^T) = \{\mathbf{y} \in \mathbb{Z}^m \mid \exists s \in \mathbb{Z}^n,\ s.t.\ \mathbf{y} \equiv \mathbf{B}^T\mathbf{s} \pmod q\}.$$

Alwen *et al.* have given an efficient algorithm to generate a random lattice along with its trapdoor basis:

**Theorem 1** [3]**.** *There is a $\mathcal{P}.\mathcal{P}.\mathcal{T}$ algorithm* TrapSamp *that, on input $1^n, 1^m, q$, with $q \geqslant 2$ and $m \geqslant 8n\log q$, outputs $(\mathbf{A}, \mathbf{T}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}^{m\times m}$ such that the distribution on $\mathbf{A}$ is statistically close to $U(\mathbb{Z}_q^{n\times m})$, and with probability all but negligible in $n$:*

1. *the columns of $\mathbf{T}$ form a basis of the lattice $\mathcal{L}^{\perp}(\mathbf{A})$;*
2. *$\|\mathbf{T}\| = O(n\log q)$ and $\|\widetilde{\mathbf{T}}\| = O(\sqrt{n\log q})$.*

Gentry, Peikert and Vaikuntanathan [10] focused on the applications of such short bases. Specifically, taking $q = poly(n), m \geqslant 8n\log q, s = \omega(\sqrt{n\log q\log n})$, a family of one-way preimage-sampleable functions is defined in the following:

1. $\mathsf{GPVGen}(1^n)$:
   (1) run $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{TrapSamp}(1^n, 1^m, q)$;
   (2) define the function: $f_{\mathbf{A}}(\mathbf{e}) = \mathbf{A}\mathbf{e} \pmod q$, with domain $\{\mathbf{e} \in \mathbb{Z}^m \mid \|\mathbf{e}\| \leqslant s\sqrt{m}\}$ and range $\mathbb{Z}_q^n$.
2. $\mathsf{SampleISIS}(\mathbf{A}, \mathbf{T}, s, \mathbf{u})$:
   (1) compute some $\mathbf{t} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{t} \equiv \mathbf{u} \pmod q$ using standard linear algebra;
   (2) sample $\mathbf{e} \leftarrow D_{\mathcal{L}^{\perp}(\mathbf{A})+\mathbf{t},s}$ using the trapdoor basis $\mathbf{T}$.

The above function is one-way if $\mathsf{GapSVP}_{\gamma}$ is hard in the worst case for polynomial approximation factor $\gamma$ [3].

**Theorem 2** [12]**.** *There is a $\mathcal{P}.\mathcal{P}.\mathcal{T}$ algorithm* SuperSamp *that, on input $1^n, 1^m, q$, with $q \geqslant 2$ and $m \geqslant n + 8n\log q$, and $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ whose columns span $\mathbb{Z}_q^n$, outputs $(\mathbf{A}, \mathbf{T}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}^{m\times m}$ such that $\mathbf{A}\mathbf{B}^T = 0 \pmod q$ and the distribution on $\mathbf{A}$ is statistically close to uniform over $\mathbb{Z}_q^{n\times m}$ subject to this condition. Moreover, with probability all but negligible in $n$:*

1. *the columns of $\mathbf{T}$ form a basis of the lattice $\mathcal{L}^{\perp}(\mathbf{A})$;*
2. *$\|\widetilde{\mathbf{T}}\| = O(\log n \cdot \sqrt{mn\log q})$.*

We now describe the $\mathsf{LWE}$ problem. Fix a positive integer $n$, integers $m \geqslant n$ and $q \geqslant 2$, a vector $\mathbf{s} \in \mathbb{Z}_q^n$, and a probability distribution $\chi$ on $\mathbb{R}^m$. Define the following two distributions over $\mathbb{Z}_q^{n\times m} \times [0,q)^m$ :

1. $\mathsf{LWE}_{m,q,\chi}$ is the distribution obtained by choosing uniform $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, sampling $\mathbf{e} \leftarrow \chi$, and outputting $(\mathbf{A}, \mathbf{A}^T\mathbf{s} + \mathbf{e} \pmod q)$.
2. $\mathsf{U}_{m,q}$ is the distribution obtained by choosing uniform $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and uniform $\mathbf{y} \in [0,q)^m$, and outputting $(\mathbf{A}, \mathbf{y})$.

Formally, for $m, q$ and $\chi$ that may depend on $n$, we say the $\mathsf{LWE}_{m,q,\chi}$ problem is hard, if the following is negligible for any $\mathcal{P}.\mathcal{P}.\mathcal{T}$ algorithm $\mathcal{D}$:

$$|Pr[\mathbf{s} \leftarrow \mathbb{Z}_q^n; (\mathbf{A}, \mathbf{y}) \leftarrow \mathsf{LWE}_{m,q,\chi}(\mathbf{s}) : \mathcal{D}(\mathbf{A}, \mathbf{y}) = 1] - Pr[(\mathbf{A}, \mathbf{y}) \leftarrow \mathsf{U}_{m,q} : \mathcal{D}(\mathbf{A}, \mathbf{y}) = 1]|$$

The error distribution $\chi$ we will use in this paper is the discrete Gaussian distribution $D_{\mathbb{Z}^m,\alpha q}$. We write $\mathsf{LWE}_{m,q,\alpha}(\mathbf{s})$ as an abbreviation for $\mathsf{LWE}_{m,q,D_{\alpha q}}(\mathbf{s})$, and $\widehat{\mathsf{LWE}}_{m,q,\alpha}(\mathbf{s})$ for $\mathsf{LWE}_{m,q,D_{\mathbb{Z}^m,\alpha q}}(\mathbf{s})$, where $D_{\alpha q}$ is the continuous Gaussian distribution.

**Lemma 1** [12]. *For any $m = m(n), q = q(n), \alpha = \alpha(n)$ satisfying $\alpha q = \omega(\sqrt{\log n})$, hardness of the $\mathsf{LWE}_{m,q,\alpha}$ problem implies hardness of the $\widehat{\mathsf{LWE}}_{m,q,\alpha\sqrt{2}}$ problem.*

## 3 A Fully Anonymous Group Signature with Verifier-Local Revocation

In this section, we present a lattice-based *group signature scheme with verifier-local revocation* (GS-VLR) holding full-anonymity and traceability as defined by a variant of the GKV scheme. Note that a lattice-based GS-VLR scheme has already been proposed by Langlois *et al.* [15] at PKC 2014. However, that scheme only holds some weaker insider-anonymity, and cannot be used in initializing our semi-generic construction. Now, we begin with recalling the notion of GS-VLR [7].

### 3.1 Group Signatures with Verifier-Local Revocation

Formally, a GS-VLR scheme $\mathcal{GS} = (\mathsf{GKg}, \mathsf{GSig}, \mathsf{GVf})$ is a tuple of three poly-time algorithms:

1. $\mathsf{GKg}(1^n, 1^N)$. The randomized key generation algorithm takes as input the security parameter $n \in \mathbb{N}$, and the group size $N \in \mathbb{N}$. It outputs a group public key $gpk$, all members' secret keys $gsk := (gsk_1, \cdots, gsk_N)$, and all members' revocation tokens $grt := (grt_1, \cdots, grt_N)$.
2. $\mathsf{GSig}(gpk, gsk_i, m)$. The randomized signing algorithm takes as input the group public key $gpk$, the signing key $gsk_i$ of member $i \in [N]$, and a message $m \in \{0,1\}^*$. It outputs a signature $\sigma$.

3. $\mathsf{GVf}(gpk, RL, m, \sigma)$. The deterministic verification algorithm takes as input the group public key $gpk$, a set of revocation tokens $RL$, a message $m$, and a candidate signature $\sigma$. It returns either 1 or 0. The latter indicates that either $\sigma$ is not a valid signature, or the member who generated it has been revoked.

**Correctness.** For all $n, N \in \mathbb{N}$, all $(gpk, grt, gsk) \leftarrow [\mathsf{GKg}(1^n, 1^N)]$, all $i \in [N]$, and all $m \in \{0,1\}^*$, the following holds with overwhelming probability:

$$\mathsf{GVf}(gpk, RL, m, \mathsf{GSig}(gpk, gsk_i, m)) = 1 \iff grt_i \notin RL.$$

**Implicit Tracing Algorithm.** Given a valid message-signature pair $(m, \sigma)$ with respect to some revocation list $RL$, the one possessing all revocation tokens $grt$ traces by the following algorithm: for $i \in [N]$, run the verification algorithm $\mathsf{GVf}(gpk, RL_i := \{grt_i\}, m, \sigma)$, and output the first index for which the verification algorithm outputs 0; otherwise output a symbol $\perp$. Apparently, if a GS-VLR scheme is correct, so is the above implicit tracing algorithm (the honestly generated signature will always trace to its originator).

**Oracles.** To formalize the security properties, the following oracles are used:

1. **GSig**$(\cdot,\cdot)$: for queries $(m, i) \in \{0,1\}^* \times [N]$, it returns $\sigma \leftarrow \mathsf{GSig}(gpk, gsk_i, m)$.
2. **Corrupt**$(\cdot)$: a corrupt set $\mathcal{C}$ is initialized as empty; for queries $i \in [N]$, it responds with $gsk_i$, and add $i$ into $\mathcal{C}$.
3. **Revoke**$(\cdot)$: for queries $i \in [N]$, it answers with $grt_i$.

**Anonymity.** In the *insider-anonymity* experiment as shown in Fig. 1, the adversary's goal is to determine which of two keys generated a signature. He is not given access to *either key or revocation token*. The deprivation of the challenge revocation tokens is necessary, if the GS-VLR scheme is correct. In contrast, the deprivation of the challenge signing keys may not be a must.

Actually, in the constructions proposed by Boneh *et al.* [7], the revocation token $grt_i$ of some member $i$ can be derived from its secret signing key $gsk_i$. This is an admirable feature on aspect of efficiency, however leaving room for improving security. Specifically, a stronger version of anonymity for GS-VLR, called *full-anonymity* can be considered.

**Definition 6 (Full-Anonymity).** *A GS-VLR scheme $\mathcal{GS}$ holds full-anonymity if for all $\mathcal{P.P.T}$ adversaries $\mathcal{A}$, the following advantage function is negligible in $n$:*

$$Adv^{GS-VLR-full-anonymity}_{\mathcal{A},\mathcal{GS}}(n) = |\Pr[1 \leftarrow \mathbf{Expt}^{GS-VLR-full-anonymity}_{\mathcal{A},\mathcal{GS}}(1^n, N)] - \frac{1}{2}|.$$

The full-anonymity differs from the insider-version in that the adversary as shown in Fig. 2 is equipped with all members' secret signing keys.

$\mathbf{Expt}^{GS-VLR-insider-anonymity}_{\mathcal{A},\mathcal{GS}}(1^n, N)$

$(gpk, grt, gsk) \leftarrow \mathsf{GKg}(1^n, 1^N)$

$(i_0, i_1, m, st) \leftarrow \mathcal{A}^{\mathbf{GSig}(\cdot,\cdot),\mathbf{Corrupt}(\cdot),\mathbf{Revoke}(\cdot)}(gpk)$

$b \xleftarrow{\$} \{0,1\}; \sigma \leftarrow \mathsf{GSig}(gpk, gsk_{i_b}, m)$

$b' \leftarrow \mathcal{A}^{\mathbf{GSig}(\cdot,\cdot)}(st, \sigma)$

**return** 1 **if** :

$b == b'$

$i_0, i_1 \notin \mathcal{C}$ and were never queried to **Revoke**$(\cdot)$

**else return** 0

**Fig. 1.** Insider-anonymity for GS-VLR.

$\mathbf{Expt}^{GS-VLR-full-anonymity}_{\mathcal{A},\mathcal{GS}}(1^n, N)$

$(gpk, grt, gsk) \leftarrow \mathsf{GKg}(1^n, 1^N)$

$(i_0, i_1, m, st) \leftarrow \mathcal{A}^{\mathbf{Revoke}(\cdot)}(gpk, gsk)$

$b \xleftarrow{\$} \{0,1\}; \sigma \leftarrow \mathsf{GSig}(gpk, gsk_{i_b}, m)$

$b' \leftarrow \mathcal{A}(st, \sigma)$

**return** 1 **if** :

$b == b'$

$i_0, i_1$ were never queried to **Revoke**$(\cdot)$

**else return** 0

**Fig. 2.** Full-anonymity for GS-VLR.

**Traceability.** In the traceability experiment as shown in Fig. 3, the adversary's goal is to forge a signature that cannot be traced to one of unrevoked malicious members in his coalition using the implicit tracing algorithm.

$\mathbf{Expt}^{GS-VLR-traceability}_{\mathcal{A},\mathcal{GS}}(1^n, N)$

$(gpk, grt, gsk) \leftarrow \mathsf{GKg}(1^n, 1^N)$

$(m, \sigma, RL) \leftarrow \mathcal{A}^{\mathbf{Corrupt}(\cdot),\mathbf{GSig}(\cdot,\cdot)}(gpk, grt)$

**return** 1 **if** :

$\mathsf{GVf}(gpk, RL, m, \sigma) = 1$

$\sigma$ traces to someone out of $\mathcal{C} \backslash RL$, or the implicit tracing fails

$\mathbf{GSig}(m, i)$ was never queried for $i \notin \mathcal{C}$

**else return** 0

**Fig. 3.** Traceability for GS-VLR.

**Definition 7 (Traceability).** *A GS-VLR scheme $\mathcal{GS}$ holds traceability if for all $\mathcal{P}.\mathcal{P}.\mathcal{T}$ adversaries $\mathcal{A}$, the following advantage function is negligible in $n$:*

$$Adv^{GS-VLR-traceability}_{\mathcal{A},\mathcal{GS}}(n) = \Pr[1 \leftarrow \mathbf{Expt}^{GS-VLR-traceability}_{\mathcal{A},\mathcal{GS}}(1^n, N)].$$

Note that if the verification algorithm satisfies that $\mathsf{GVf}(gpk, RL, m, \sigma) = 1 \iff \mathsf{GVf}(gpk, \{grt_i\}, m, \sigma) = 1$ for all $grt_i \in RL$, the success condition that $\sigma$ traces to someone out of $\mathcal{C} \backslash RL$ can be equivalently modified as that $\sigma$ traces to someone out of $\mathcal{C}$. This is because $\sigma$ will never be traced to some $i^* \in (\mathcal{C} \bigcap RL)$, otherwise by the definition of the implicit tracing algorithm, we have $\mathsf{GVf}(gpk, \{grt_{i^*}\}, m, \sigma) = 0$ which contradicts with $\mathsf{GVf}(gpk, \{grt_{i^*}\}, m, \sigma) = 1$.

### 3.2 A Fully Anonymous GS-VLR Scheme from Lattice Assumptions

Let $n$ be the security parameter, $q = poly(n), m \geqslant 8n \log q$ and $s \geqslant C\sqrt{n \log q} \cdot \omega(\sqrt{\log m})$ be parameters of the system. Let $H : \{0,1\}^* \to \mathbb{Z}_q^n$ be a hash function, to be modeled as a random oracle. The GS-VLR scheme is demonstrated as follow:

1. $\mathsf{GKg}(1^n, 1^N)$: for $i \in [N]$, compute $(\mathbf{B}_i, \mathbf{S}_i) \leftarrow \mathsf{TrapSamp}(1^n, 1^m, q), (\mathbf{A}_i, \mathbf{T}_i) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{B}_i)$. Output $gpk := ((\mathbf{A}_i, \mathbf{B}_i))_{i=1}^N$ as the group public key, $gsk := (\mathbf{T}_i)_{i=1}^N$ as members' signing keys, $grt := (\mathbf{S}_i)_{i=1}^N$ as members' revocation tokens.
2. $\mathsf{GSig}(gpk, gsk_i, m)$: it works exactly the same as the signing algorithm of the GKV scheme.
3. $\mathsf{GVf}(gpk, RL, m, \sigma)$: parse the signature $\sigma$ as $(\mathbf{r}, \mathbf{c}_1, \cdots, \mathbf{c}_N, \pi)$. If $\pi$ is not valid, output 0; for $i \in [N]$, calculate $\mathbf{h}_i \leftarrow H(m \| \mathbf{r} \| i)$, if the equation $\mathbf{A}_i \mathbf{c}_i \equiv \mathbf{h}_i \pmod q$ does not hold, output 0; for $\mathbf{S}_{i_\ell} \in RL$ $(\ell = 1, \cdots, |RL|)$, calculate $\mathbf{e}'_{i_\ell} \leftarrow \mathbf{S}_{i_\ell}^T \mathbf{c}_{i_\ell} \pmod q, \mathbf{e}_{i_\ell} \leftarrow \mathbf{S}_{i_\ell}^{T^{-1}} \mathbf{e}'_{i_\ell}$, and if $\|\mathbf{e}_{i_\ell}\| \leqslant s\sqrt{m}$, output 0. Otherwise output 1.

The only difference between the GKV scheme and ours lies in the verification procedure. Namely, we incorporate the original GKV verification algorithm and open algorithm into our new verification algorithm.

For the correctness and security of our scheme, we have the following theorems.

**Theorem 3.** *Our GS-VLR scheme is correct.*

**Theorem 4.** *Let $m, q, s$ be described as above. If $\mathsf{LWE}_{m,q,\alpha}$ is hard for $\alpha = s/(q\sqrt{2})$, and $\mathsf{GapSVP}_\gamma$ is hard for $\gamma = O(n \log^4 n)$, and the proof system used is witness-indistinguishable, our GS-VLR scheme is fully anonymous and traceable.*

## 4 Hierarchical Group Signatures with Verifier-Local Revocation

In this section, we formalize the new notion of *hierarchical group signatures with verifier-local revocation* (HGS-VLR), and propose a semi-generic construction from GS-VLR.

### 4.1 Syntax and Correctness

We follow the notations from [26]. There are two types of parties: *signers* denoted as $S_\alpha$ for $\alpha$ in some index set $\mathcal{I}$ and *managers* denoted as $M_\alpha$ for indices $\alpha$ described below. If a manager directly manages a set of signers $\{\alpha \mid \alpha \in \beta \subset \mathcal{I}\}$, we denote it by $M_\beta$; if a manager directly manages a set of managers $\{M_{\beta_1}, \cdots, M_{\beta_\ell}\}$, we denote it by $M_\gamma$ where $\gamma = \{\beta_1, \cdots, \beta_\ell\}$.

All parties are organized in a balanced tree $\mathcal{T}$ of depth $\delta \in \mathbb{N}$, where signers are leaves and managers are inner nodes. For $i \in \{0, \cdots, \delta\}$, let $\mathcal{T}^i$ denote all nodes at depth $i$; we denote all leaves by $\mathcal{L}(\mathcal{T})$ and the root by $\rho$. When there is no risk of confusion, we write $\alpha$ instead of $M_\alpha$ or $S_\alpha$.

An HGS-VLR scheme $\mathcal{HGS} = (\mathsf{HKg}, \mathsf{HSig}, \mathsf{HVf})$ consists of three poly-time algorithms:

1. $\mathsf{HKg}(1^n, \mathcal{T})$. The randomized key generation algorithm takes as input the security parameter $n \in \mathbb{N}$, and a balanced tree $\mathcal{T}$ of size polynomially bounded in $n$. It outputs a tuple of maps $(hpk, hrt, hsk)$, where $hpk$ and $hrt$ associates each node $\alpha \in \mathcal{T}$ with a public value $hpk(\alpha)$ and a revocation token $hrt(\alpha)$, and $hsk$ associates each leaf $\alpha \in \mathcal{L}(\mathcal{T})$ with a secret signing key $hsk(\alpha)$.
2. $\mathsf{HSig}(hpk(\mathcal{T}), hsk(\alpha), m)$. The randomized signing algorithm takes as input the public map $hpk(\mathcal{T})$, a message $m \in \{0,1\}^*$, and the secret signing key $hsk(\alpha)$ of some signer $\alpha \in \mathcal{L}(\mathcal{T})$, and returns a signature $\sigma$.
3. $\mathsf{HVf}(hpk(\mathcal{T}), RL, m, \sigma)$. The deterministic verification algorithm takes as input the public map $hpk(\mathcal{T})$, a revocation list $RL \subset hrt(\mathcal{T})$ composed of the tokens associating with already revoked parties, a message $m$, and a candidate signature $\sigma$. It returns either 1 or 0, and the latter means either that $\sigma$ is not a valid signature, or (at least) one on the path from the signer to the root (both are included) has been revoked.

The key generation algorithm $\mathsf{HKg}$ is run by some trusted key generator $\mathcal{TKG}$, akin to the circumstance in HGS. The map $hpk(\mathcal{T})$ is made public, and each signer $\alpha \in \mathcal{L}(\mathcal{T})$ is given its secret signing key $hsk(\alpha)$. We initialize the public revocation list $RL$ as empty, and any party $\alpha \in \mathcal{T}$ can be revoked by simply adding its revocation token $hrt(\alpha)$ into $RL$.

**Correctness.** An HGS-VLR scheme is *correct*, if for all $n \in \mathbb{N}$, all balanced trees $\mathcal{T}$ of depth $\delta \in \mathbb{N}$ and size polynomially bounded in $n$, all $(hpk, hrt, hsk) \in [\mathsf{HKg}(1^n, \mathcal{T})]$, all $m \in \{0,1\}^*$, and all $\alpha \in \mathcal{L}(\mathcal{T})$, the following holds with overwhelming probability:

$$\mathsf{HVf}(hpk(\mathcal{T}), RL, m, \mathsf{HSig}(hpk(\mathcal{T}), hsk(\alpha), m)) = 1 \iff \{hrt(\gamma)\}_{\gamma \in Ancestor(\alpha)} \bigcap RL = \phi,$$

where $Ancestor(\alpha)$ denotes all nodes on the path from $\alpha$ to $\rho$ with both included. We highlight that, if some manager $\beta \in (\mathcal{T} - \mathcal{L}(\mathcal{T}))$ is ever revoked by adding $hrt(\beta)$ into $RL$, all signers whom $\beta$ (maybe indirectly) manages are no longer valid, even without adding their tokens into $RL$ explicitly.

**Implicit Tracing Algorithm.** Each manager $\beta \in (\mathcal{T} - \mathcal{L}(\mathcal{T}))$ is given all revocation tokens $\{hrt(\gamma)\}_{\gamma \in \beta}$ of its direct children, to run an implicit tracing algorithm inherent to HGS-VLR. Specifically, given a valid message-signature pair $(m, \sigma)$, a manager $\beta$ does the following:

1. For $\gamma \in \beta$, run $\mathsf{HVf}(hpk(\mathcal{T}), RL_\gamma := \{hrt(\gamma)\}, m, \sigma)$;
2. Output the first index for which the verification algorithm says 0 and terminate; if the pair $(m, \sigma)$ passes all verifications, output a symbol $\perp$.

This algorithm is correct, if the HGS-VLR scheme is correct as defined. Namely, given a valid message-signature pair $(m, \sigma)$ honestly generated by some signer $\alpha \in \mathcal{L}(\mathcal{T})$, *a path-following tracing*, where some father node (initialized as the root) traces then passes the pair to some child, will always locate the signer $\alpha$ eventually.

To explain, let $\beta_0 := \rho \ni \beta_1 \ni \cdots \ni \beta_\delta := \alpha$ be the path. If the HGS-VLR scheme is correct, then by definition $\mathsf{HVf}(hpk(\mathcal{T}), RL, m, \sigma) = 1 \iff \{hrt(\beta_i)\}_{i=0}^{\delta} \bigcap RL = \phi$ holds with overwhelming probability; it follows that with overwhelming probability, $\mathsf{HVf}(hpk(\mathcal{T}), \{hrt(\beta)\}, m, \sigma) = 0 \iff \beta = \beta_i$ for some $i \in \{0, 1, \cdots, \delta\}$. Then for $i$ from 0 to $\delta - 1$, the manager $\beta_i$ will always trace $(m, \sigma)$ to $\beta_{i+1}$ independently with probability $(1 - negl(n))$, where $negl(n)$ is a negligible function with respect to $n$. Then the probability of locating $\alpha$ is $(1 - negl(n))^\delta$, which is still overwhelming since $\delta$ is polynomial in $n$. [1]

### 4.2 Security Model

We formalize two security requirements for HGS-VLR, namely *full-anonymity* and *traceability* using the experiments as shown in Fig. 4 and Fig. 5 respectively. Overall, we use the framework of Bellare *et al.*, and thus these two properties are strong enough to capture all related security requirements, e.g., unforgeability, exculpability, collision resistance, framing, unlinkability and so on as argued by Bellare *et al.* [5]. To begin with, we specify the following oracles:

1. **HCorrupt**$(\cdot)$: a corrupt set $\mathcal{C}$ is initialized as empty; for queries $\alpha \in \mathcal{L}(\mathcal{T})$, it responds with $hsk(\alpha)$, and adds $\alpha$ into the set $\mathcal{C}$.
2. **HRevoke**$(\cdot)$: for queries $\alpha \in \mathcal{T}$, it returns $hrt(\alpha)$.
3. **HSig**$(\cdot, \cdot)$: for queries $(m, \alpha) \in \{0,1\}^* \times \mathcal{L}(\mathcal{T})$, it returns $\sigma \leftarrow \mathsf{HSig}(hpk(\mathcal{T}), hsk(\alpha), m)$.

**Full-Anonymity.** Assume that a message $m$ has been signed by either $\alpha^{(0)}$ or $\alpha^{(1)} \in \mathcal{L}(\mathcal{T})$. Let $B$ denote all nodes on paths from $\alpha^{(0)}$ and $\alpha^{(1)}$ up to their first common ancestor $\alpha_t$, including $\alpha^{(0)}$ and $\alpha^{(1)}$ but excluding $\alpha_t$. One having access to arbitrary element of $\{hrt(\beta)\}_{\beta \in B}$ can trivially determine who the signer is, if the HGS-VLR scheme is correct. Full-anonymity says that nobody can determine whether $\alpha^{(0)}$ or $\alpha^{(1)}$ signed the message without access to $\{hrt(\beta)\}_{\beta \in B}$, even if it is given all secret signing keys $hsk(\mathcal{L}(\mathcal{T}))$, and is allowed to select the challenge identities $\alpha^{(0)}$ and $\alpha^{(1)}$ as well as the challenge message $m$ by itself.

Note that no generality is lost by having access to the oracle **HRevoke** only before $\sigma$ is computed, since $\mathcal{A}$ has decided on $\alpha^{(0)}$ and $\alpha^{(1)}$ and can obtain any $hrt(\alpha)$ with $\alpha \notin B$ before it receives $\sigma$. When $\mathcal{T}$ is a depth one tree, the experiment in Fig. 4 reduces to the experiment in Fig. 2 for GS-VLR.

[1] The negligible functions might be different for different $\beta_i$, however, we express them uniformly by $nelg(n)$ for simplicity, since both lead to a overwhelming probability.

$\mathbf{Expt}^{HGS\text{-}VLR\text{-}full\text{-}anonymity}_{\mathcal{A},\mathcal{HGS}}(1^n, \mathcal{T})$

$(hpk, hrt, hsk) \leftarrow \mathsf{HKg}(1^n, \mathcal{T})$

$(\alpha^{(0)}, \alpha^{(1)}, m, st) \leftarrow \mathcal{A}^{\mathbf{HRevoke}(\cdot)}(hpk(\mathcal{T}), hsk(\mathcal{L}(\mathcal{T})))$

$b \xleftarrow{\$} \{0,1\}; \sigma \leftarrow \mathsf{HSig}(hpk(\mathcal{T}), hsk(\alpha^{(b)}), m)$

$b' \leftarrow \mathcal{A}(st, \sigma)$

**return** 1 **if** :

$b == b'$

**HRevoke**($\beta$) was never queried for $\beta \in B$

**else return** 0

**Fig. 4.** Full-anonymity for HGS-VLR.

**Definition 8 (Full-Anonymity).** *An HGS-VLR scheme holds full-anonymity if for all $\mathcal{P}.\mathcal{P}.\mathcal{T}$ adversaries $\mathcal{A}$, the following advantage function is negligible in $n$:*

$$Adv^{HGS-VLR-full-anonymity}_{\mathcal{A},\mathcal{HGS}}(n) = |\Pr[1 \leftarrow \mathbf{Expt}^{HGS-VLR-full-anonymity}_{\mathcal{A},\mathcal{HGS}}(1^n, \mathcal{T})] - \frac{1}{2}|.$$

**Traceability.** In the experiment as shown in Fig. 5, the adversary $\mathcal{A}$ is provided with all revocation tokens $hrt(\mathcal{T})$ and allowed to adaptively corrupt a coalition of signers (denoted by $\mathcal{C}$). To win, $\mathcal{A}$ must come up with a valid message-signature pair $(m, \sigma)$ with respect to some revocation list $RL$, and one of the following must hold: ($a$) *all* managers *at the penultimate depth* cannot figure out an identity by the implicit tracing algorithm; or ($b$) there exists *one* manager *at the penultimate depth* tracing to someone not in the coalition $\mathcal{C} \backslash RL$. For simplicity, let $O$ denote the open results of all managers at the penultimate depth, and we formalize ($a$) and ($b$) into two expressions, namely $O = \{\perp\}$ and $O \bigcap (\mathcal{L}(\mathcal{T}) - \mathcal{C} \backslash RL) \neq \phi$ respectively.

Now we demonstrate the implications. Given some valid message-signature pair with respect to some revocation list, it is always feasible to figure out the actual (and unrevoked) generator while nobody will be framed. Specifically, if the *path-following tracing* does not fail, it will always locate the actual (and unrevoked) generator; if the path-following tracing fails somewhere, all managers at the penultimate depth can do a *whole-depth tracing*, and figure out some unrevoked malicious signer, while no honest signer will be traced.

Overall, when the path-following tracing fails, attacks against traceability are detected, or contradicting with the correctness; on the other hand, the whole-depth tracing stands for the capability to find out the attackers. Honestly, the latter is less efficient than the former, but it will merely be used if the punishment of misbehaving is heavily enough (this seems rather sound in the context of anonymous credit card systems). Note that such *detect-and-punish* paradigm is widely used in the e-cash setting, e.g., in solving the double-spending problem [24].

$\mathbf{Expt}^{HGS\text{-}VLR\text{-}traceability}_{\mathcal{A},\mathcal{HGS}}(1^n, \mathcal{T})$

1 : $(hpk, hrt, hsk) \leftarrow \mathsf{HKg}(1^n, \mathcal{T})$
2 : $(m, \sigma, RL) \leftarrow \mathcal{A}^{\mathbf{HCorrupt}(\cdot),\mathbf{HSig}(\cdot,\cdot)}(hpk(\mathcal{T}), hrt(\mathcal{T}))$
3 : **return** 1 **if** :
4 : $\quad \mathsf{HVf}(hpk(\mathcal{T}), RL, m, \sigma) = 1$
5 : $\quad O \bigcap (\mathcal{L}(\mathcal{T}) - \mathcal{C} \backslash RL) \neq \phi$, **or** $O = \{\perp\}$
6 : $\quad$ **HSig**$(m, \alpha)$ was never queried for $\alpha \notin \mathcal{C}$
7 : **else return** 0

**Fig. 5.** Traceability for HGS-VLR.

The experiment above reduces to the experiment in Fig. 3 for GS-VLR, when $\mathcal{T}$ is a depth one tree. Moreover, when nobody is corrupted, namely $\mathcal{C} = \phi$, the requirement of *unforgeability* is reflected.

**Definition 9 (Traceability).** *An HGS-VLR scheme holds traceability if for all $\mathcal{P}.\mathcal{P}.\mathcal{T}$ adversaries $\mathcal{A}$, the following advantage function is negligible in $n$:*

$$Adv^{HGS-VLR-traceability}_{\mathcal{A},\mathcal{HGS}}(n) = \Pr[1 \leftarrow \mathbf{Expt}^{HGS-VLR-traceability}_{\mathcal{A},\mathcal{HGS}}(1^n, \mathcal{T})].$$

### 4.3 A Semi-generic HGS-VLR Construction

As we will see, the construction is quite natural under our model, but definitely nontrivial. First, we reduce the full-anonymity of HGS-VLR to that of the underlying GS-VLR; however, similar reduction is not right when it comes to the insider-anonymity counterpart (we can similarly define insider-anonymity for HGS-VLR). Second, it essentially reflects a different designing paradigm from Wikström *et al.*'s, which adds the capability of revocation to the original notion. Now we show the semi-generic construction of HGS-VLR from GS-VLR.

Specifically, let $\mathcal{T}$ be some balanced tree of depth $\delta$, and let $\mathcal{GS} = (\mathsf{GKg}, \mathsf{GSig}, \mathsf{GVf})$ be the underlying GS-VLR. We construct the HGS-VLR scheme as following:

1. $\mathsf{HKg}(1^n, \mathcal{T})$: for $i \in \{0, \cdots, \delta\}$, run $(gpk^i, grt^i, gsk^i) \leftarrow \mathsf{GKg}(1^n, 1^{|\mathcal{T}^i|})$, where $grt^i := \{grt_\beta\}_{\beta \in \mathcal{T}^i}$ and $gsk^i := \{gsk_\beta\}_{\beta \in \mathcal{T}^i}$. The public map $hpk$ is defined as: $hpk(\rho) = \{gpk^i\}_{i=0}^{\delta}$, $hpk(\alpha) = \phi$ for $\alpha \in (\mathcal{T} - \rho)$; the secret map $hsk$ is defined as: $hsk(\alpha) = \{gsk_\gamma\}_{\gamma \in Ancestor(\alpha)}$ for $\alpha \in \mathcal{L}(\mathcal{T})$, where $Ancestor(\alpha)$ denotes all nodes on the path from $\alpha$ to $\rho$ with both included; $hrt(\beta) = grt_\beta$ for $\beta \in \mathcal{T}$.
2. $\mathsf{HSig}(hpk(\mathcal{T}), hsk(\alpha), m)$: for $i \in \{0, \cdots, \delta\}$, run $\sigma_i \leftarrow \mathsf{GSig}(gpk^i, gsk_\gamma, m)$, where $\{\gamma\} = Ancestor(\alpha) \bigcap \mathcal{T}^i$, and output the signature $\sigma := (\sigma_0, \cdots, \sigma_\delta)$.

3. $\mathsf{HVf}(hpk(\mathcal{T}), RL, m, \sigma)$: parse the candidate signature $\sigma$ as $(\sigma_0, \cdots, \sigma_\delta)$. For $i \in \{0, \cdots, \delta\}$, run $\mathsf{GVf}(gpk^i, RL_i, m, \sigma_i)$, where $RL_i := \{grt_\beta \in RL \mid \beta \in \mathcal{T}^i\}$ denotes all revocation tokens in RL associating with parties at depth $i$, and if $\mathsf{GVf}(gpk^i, RL_i, m, \sigma_i) = 0$, terminate and output 0; if $(m, \sigma)$ passes all verifications, output 1.

For the integrity of the revocation functionality, we generate both token and signing key for the root $\rho$. In other words, $hrt(\rho)$ exists only for revocation, while other tokens are used either in the implicit tracing (thus a manager $\beta$ is given all children's tokens $\{hrt(\gamma)\}_{\gamma \in \beta}$), or removing someone out of the hierarchy by adding its token into $RL$. If there is no need for revoking the root, the scheme can be modified by simply dropping the group composed of the root.

For the correctness and security of our construction, we have the following theorems.

**Theorem 5.** *If the underlying GS-VLR is correct, the HGS-VLR scheme resulted from our semi-generic construction is also correct.*

**Theorem 6.** *The HGS-VLR scheme described above holds full-anonymity and traceability, if the underlying GS-VLR scheme holds full-anonymity and traceability.*

When we instantiate the semi-generic construction by the variant of the GKV scheme as shown in Sect. 3.2, a concrete construction from lattice assumptions is given. Overall, we regard all parties at the same depth as a group respectively. As for the HGS-VLR scheme, its anonymity is reduced to the anonymity of all groups; however, its traceability is reduced to the traceability of the group composed of all leaves. This leaves rooms for improving efficiency, in the meaning that for the first $(\delta - 1)$ groups, the underlying scheme may only holds anonymity and correctness.

## 5 Summary

The significance of this paper is embodied on two aspects: first, from HGS to HGS-VLR, we provide the former with efficient revocation approach by introducing and formalizing the latter new notion; second, in contrast with the generic HGS construction, we do not employ extra building blocks (e.g., an anonymous encryption scheme [11]) in our semi-generic HGS-VLR construction, and this somehow unifies the studies of HGS-VLR and GS-VLR. The expansion of signature's size is in proportion to the tree's depth. However, this won't bother much since the depth is usually small in applications. For future works, on one hand, it is desirable to depict a fully dynamic case by further adding the capability of dynamic joining; on the other hand, constructing efficient schemes in the lattice setting seems to be some long-term open problem, as reflected in the studies of GS.

**Acknowledgement.** The authors would like to thank the anonymous reviewers of ICICS 2018 for helpful comments.

## References


1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, STOC 1996, pp. 99–108. ACM, New York. https://doi.org/10.1145/237814.237838
2. Ajtai, M.: The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, pp. 10–19. STOC 1998. ACM, New York (1998). https://doi.org/10.1145/276698.276705
3. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. Theory Comput. Syst. **48**(3), 535–553 (2011). https://doi.org/10.1007/s00224-010-9278-3
4. Ateniese, G., Camenisch, J., Joye, M., Tsudik, G.: A practical and provably secure coalition-resistant group signature scheme. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 255–270. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44598-6_16
5. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_38
6. Bender, J., Dagdelen, Ö., Fischlin, M., Kügler, D.: Domain-specific pseudonymous signatures for the german identity card. In: Gollmann, D., Freiling, F.C. (eds.) Information Security, pp. 104–119. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33383-5_7
7. Boneh, D., Shacham, H.: Group signatures with verifier-local revocation. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, pp. 168–177. ACM, New York (2004). https://doi.org/10.1145/1030083.1030106
8. Camenisch, J., Neven, G., Rückert, M.: Fully anonymous attribute tokens from lattices. In: Visconti, I., De Prisco, R. (eds.) Security and Cryptography for Networks, pp. 57–75. Springer, Berlin Heidelberg (2012). https://doi.org/10.1007/978-3-642-32928-9_4
9. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-46416-6_22
10. Gentry, C., Vaikuntanathan, V., Peikert, C.: How to use a short basis: trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, 17–20 May 2008 (2008)
11. Goldwasser, S., Macali, S.: Probabilistic Encryption. J. Comput. Syst. Sci. **28**(2), 270–299 (1984)
12. Gordon, S.D., Katz, J., Vaikuntanathan, V.: A group signature scheme from lattice assumptions. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 395–412. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_23
13. Kiayias, A., Tsiounis, Y., Yung, M.: Traceable signatures. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 571–589. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_34
14. Laguillaumie, F., Langlois, A., Libert, B., Stehlé, D.: Lattice-based group signatures with logarithmic signature size. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8270, pp. 41–61. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42045-0_3

15. Langlois, A., Ling, S., Nguyen, K., Wang, H.: Lattice-based group signature scheme with verifier-local revocation. In: Krawczyk, H. (ed.) PKC 2014, pp. 345–361. Springer, Heidelberg (2014). https://doi.org/10.1007/s12204-017-1837-1
16. Libert, B., Ling, S., Mouhartem, F., Nguyen, K., Wang, H.: Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 373–403. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_13
17. Libert, B., Ling, S., Nguyen, K., Wang, H.: Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without trapdoors. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 1–31. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_1
18. Libert, B., Mouhartem, F., Nguyen, K.: A lattice-based group signature scheme with message-dependent opening. In: Manulis, M., Sadeghi, A.-R., Schneider, S. (eds.) ACNS 2016. LNCS, vol. 9696, pp. 137–155. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39555-5_8
19. Ling, S., Nguyen, K., Wang, H.: Group signatures from lattices: simpler, tighter, shorter, ring-based. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 427–449. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_19
20. Ling, S., Nguyen, K., Wang, H., Xu, Y.: Lattice-based group signatures: achieving full dynamicity with ease. In: Gollmann, D., Miyaji, A., Kikuchi, H. (eds.) Applied Cryptography and Network Security, pp. 293–312. Springer International Publishing, Cham (2017). https://doi.org/10.1007/978-3-319-61204-1_15
21. Ling, S., Nguyen, K., Wang, H., Xu, Y.: Constant-size group signatures fromÂ lattices. In: Abdalla, M., Dahab, R. (eds.) Public-Key Cryptography - PKC 2018, pp. 58–88. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-76581-5_3
22. Lysyanskaya, A., Ramzan, Z.: Group blind digital signatures: a scalable solution to electronic cash. In: Hirchfeld, R. (ed.) FC 1998. LNCS, vol. 1465, pp. 184–197. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055483
23. Nguyen, P.Q., Zhang, J., Zhang, Z.: Simpler efficient group signatures from lattices. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 401–426. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_18
24. Peter, W.: Digital Cash: Commerce on the Net. Academic Press, Cambridge (1996)
25. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_32
26. Trolin, M., Wikström, D.: Hierarchical group signatures. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) Automata, Languages and Programming, pp. 446–458. Springer, Berlin Heidelberg (2005). https://doi.org/10.1007/11523468_37