SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Hierarchical Group Signature with Verifier-Local Revocation Revisited

Lin HOU[1,2], Dongdai LIN[1,2*] & Renzhang LIU[3]

[1]*SKLOIS, Institute of Information Engineering, CAS, Beijing 100195, China;*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China;*
[3]*Westone Cryptologic Research Center, Westone Information Industry INC., Beijing 100070, China*

Appendix A Notations

Throughout this paper, the security parameter is denoted by $n \in \mathbb{N}$. We use *p.p.t.* for expressing "probabilistic polynomial-time". Let $[N]$ denote the set $\{1, 2, \cdots, N\}$ for $N \in \mathbb{N}$, and $[A]$ all possible outputs of algorithm A. Given two sets A and B, we denote $\{a \mid a \in A, a \notin B\}$ by $A \backslash B$, or by $A - B$ if $B \subseteq A$, and we denote the cardinality of A by $|A|$.

By convention, vectors are in column form and are denoted by bold lower-case letters, e.g., $\mathbf{x}$; matrices are written as bold capital letters, e.g., $\mathbf{X}$, and the i^{th} column of $\mathbf{X}$ is denoted by $\mathbf{x}_i$. The norm of a matrix is the norm (implicitly the Euclidean ℓ_2 norm) of its longest column: $\|\mathbf{X}\|=\max_i\|\mathbf{x}_i\|$. Standard big-$O$ notation is used to classify the growth of functions. Denote $f(n) = \widetilde{O}(g(n))$ if $f(n) = O(g(n) \cdot \log^c g(n))$ for some constant c, and denote $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $g(n) = O(f(n))$. We say a function $f(n)$ is negligible in n, denoted by $\mathsf{negl}(n)$, if $f(n) = o(n^{-c})$ for every constant c, and a probability is called overwhelming if it is $1 - \mathsf{negl}(n)$.

Appendix B Some background on lattices

First, we recall the definition of lattice:

Definition 1. A k-dimensional lattice of rank $m \leqslant k$ is defined as

$$\Lambda(\mathbf{B}) = \{\mathbf{B} \cdot \mathbf{c} \mid \mathbf{c} \in \mathbb{Z}^m\},$$

where $\mathbf{B} \in \mathbb{R}^{k \times m}$ consists of m linearly independent columns is called the basis.

For $\mathbf{A} \in \mathbb{Z}_q^{m \times k}$, we have a special class of k-dimensional lattices:

$$\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^k \mid \mathbf{A} \cdot \mathbf{x} = \mathbf{0} \pmod q\}.$$

Definition 2. Given a lattice Λ of rank m, the i^{th} $(1 \leqslant i \leqslant m)$ successive minimum $\lambda_i(\Lambda)$ is defined as the smallest $r > 0$ such that Λ contains at least i linearly independent lattice vectors of norm at most r.

Using this definition, we describe a standard *worst-case* approximation problem on lattices, namely the Shortest Independent Vectors Problem (SIVP), with the approximation factor $\gamma = \gamma(k)$ being some function of the dimension.

Definition 3. An instance of SIVP_γ is a full-rank basis $\mathbf{B}$ of a k-dimensional lattice $\Lambda = \Lambda(\mathbf{B})$, and it asks to output a set of k linearly independent lattice vectors $\mathbf{S} \subseteq \Lambda$ (view $\mathbf{S}$ as the set of its columns) such that $\|\mathbf{S}\| \leqslant \gamma \cdot \lambda_k(\Lambda)$.

Two *average-case* lattice problems are widely used in cryptographic designs, namely the homogeneous/Inhomogeneous Small Integer Solution (SIS/ISIS) problem [1] and the Learning with Error (LWE) problem [2], with hardness given by worst-case to average-case reductions [2,3] from SIVP.

Definition 4. Let χ be a probability distribution over $\mathbb{Z}$. For $\mathbf{s} \in \mathbb{Z}_q^k$, let $A_{\mathbf{s},\chi}$ denote the distribution obtained by sampling $\mathbf{a} \leftarrow \mathbb{Z}_q^k$ and $e \leftarrow \chi$, and outputting $(\mathbf{a}, \mathbf{a}^T \cdot \mathbf{s} + e) \in \mathbb{Z}_q^k \times \mathbb{Z}_q$. $\mathsf{LWE}_{k,m,q,\chi}$ asks to distinguish m samples chosen according to $A_{\mathbf{s},\chi}$ (for some $\mathbf{s} \leftarrow \mathbb{Z}_q^k$) and m samples chosen according to the uniform distribution on $\mathbb{Z}_q^k \times \mathbb{Z}_q$.

Definition 5. Given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times k}$, as well as a uniformly random vector $\mathbf{u} \in \mathbb{Z}_q^m$,

1. $\mathsf{SIS}_{m,k,q,\beta}$ asks to find a non-zero vector $\mathbf{x} \in \Lambda^{\perp}(\mathbf{A})$ such that $\|\mathbf{x}\| \leqslant \beta$;
2. $\mathsf{ISIS}_{m,k,q,\beta}$ asks to find a vector $\mathbf{x}' \in \{\mathbf{x} \in \mathbb{Z}^k \mid \mathbf{A} \cdot \mathbf{x} = \mathbf{u} \pmod q\}$ such that $\|\mathbf{x}'\| \leqslant \beta$.

[*] Corresponding author (email: ddlin@iie.ac.cn)

Appendix C The model of VLR-GS

We follow the work of Boneh and Shacham [4]. Formally, VLR-GS consists of three *p.p.t.* algorithms:

1. $\mathsf{GKg}(1^n, 1^N)$: On input the security parameter and the group size, this algorithm outputs a group public key gpk, and revocation tokens $grt := \{grt[i]\}_{i\in[N]}$ as well as signing keys $gsk := \{gsk[i]\}_{i\in[N]}$ of all group members.
2. $\mathsf{GSig}(gpk, gsk[i], m)$: The signing algorithm takes as input the group public key, the signing key of some member $i \in [N]$ and a message m, and outputs a signature σ.
3. $\mathsf{GVf}(gpk, RL, m, \sigma)$: The verification algorithm takes as input the group public key, *a public revocation list* $RL \subseteq grt$ and a candidate message-signature pair (m, σ), and it outputs a bit b; if $b = 0$, it means either (m, σ) is not valid or its signer has already been revoked.

Definition 6. A VLR-GS scheme is correct, if for any $n, N \in \mathbb{N}$, $(gpk, grt, gsk) \in [\mathsf{GKg}(1^n, 1^N)]$, $i \in [N]$ and any message m, it holds except with negligible probability that:

$$\mathsf{GVf}(gpk, RL, m, \mathsf{GSig}(gpk, gsk[i], m)) = 1 \iff grt[i] \notin RL.$$

Implicit Tracing. Note that no open algorithm is explicitly given in the original syntax of VLR-GS. This is because its verification algorithm can be used in the following manner for tracing purposes, assuming the possession of grt: given a valid message-signature pair (m, σ), run $\mathsf{GVf}(gpk, grt[i], m, \sigma)$ for $i \in [N]$ and output the first index for which the verification algorithm says 0; otherwise, output a symbol $\perp$ indicating a failed trace. It can be easily checked if a VLR-GS scheme is correct, this algorithm will always output the correct signer identity for honestly generated pairs.

Anonymity. Basically, two versions of anonymity have been defined for VLR-GS, namely full-anonymity and insider-anonymity/selfless-anonymity. As indicated by the name, they differ in whether the adversary has access to the signing keys of two challenge identities, as shown in Fig. C1 and Fig. C2 respectively. The following oracles are used in the experiments:

1. $\mathbf{GSig}(\cdot,\cdot)$: on queries $(m, i \in [N])$, this oracle returns $\sigma \leftarrow \mathsf{GSig}(gpk, gsk[i], m)$;
2. $\mathbf{CorruptK}(\cdot)$: a set $\mathcal{CK}$ is initialized as $\emptyset$; on queries $i \in [N]$, this oracle returns the signing key $gsk[i]$ of member i, and add i into $\mathcal{CK}$;
3. $\mathbf{CorruptT}(\cdot)$: a set $\mathcal{CT}$ is initialized as $\emptyset$; on queries $i \in [N]$, this oracle returns the revocation token $grt[i]$ of member i, and add i into $\mathcal{CT}$.

Definition 7. A VLR-GS scheme is called fully anonymous, if for any *p.p.t.* adversary $\mathcal{A}$, the following advantage function is negligible:

$$adv^{fa}_{\mathcal{A},\mathcal{GS}}(n) = |Pr[1 \leftarrow \mathbf{Expt}^{fa,0}_{\mathcal{A},\mathcal{GS}}(1^n, N)] - Pr[1 \leftarrow \mathbf{Expt}^{fa,1}_{\mathcal{A},\mathcal{GS}}(1^n, N)]|.$$

Definition 8. A VLR-GS scheme holds insider-anonymity, if for any *p.p.t.* adversary $\mathcal{A}$, the following advantage function is negligible:

$$adv^{ia}_{\mathcal{A},\mathcal{GS}}(n) = |Pr[1 \leftarrow \mathbf{Expt}^{ia,0}_{\mathcal{A},\mathcal{GS}}(1^n, N)] - Pr[1 \leftarrow \mathbf{Expt}^{ia,1}_{\mathcal{A},\mathcal{GS}}(1^n, N)]|.$$

For insider-anonymous VLR-GS schemes, the revocation token of a member can be easily derived from its signing key, and this is actually why the adversary cannot be given the signing keys of challenge identities. This feature is admirable in terms of storage, and it potentially eliminates the need for a trusted revocation authority.

```
$\mathbf{Expt}^{fa,b}_{\mathcal{A},\mathcal{GS}}(1^n, N)$
$(gpk, grt, gsk) \leftarrow \mathsf{GKg}(1^n, 1^N)$;
$(i_0, i_1, m) \leftarrow \mathcal{A}^{\mathbf{CorruptT}(\cdot)}(gpk, gsk)$;
$\sigma \leftarrow \mathsf{GSig}(gpk, gsk[i_b], m)$;
$b' \leftarrow \mathcal{A}(\sigma)$;
return $b'$ if :
    $\mathcal{CT} \bigcap \{i_0, i_1\} = \emptyset$;
else return 0.
```

Figure C1 Full-anonymity for VLR-GS.

```
$\mathbf{Expt}^{ia,b}_{\mathcal{A},\mathcal{GS}}(1^n, N)$
$(gpk, grt, gsk) \leftarrow \mathsf{GKg}(1^n, 1^N)$;
$(i_0, i_1, m) \leftarrow \mathcal{A}^{\mathbf{GSig}(\cdot,\cdot), \mathbf{CorruptK}(\cdot), \mathbf{CorruptT}(\cdot)}(gpk)$;
$\sigma \leftarrow \mathsf{GSig}(gpk, gsk[i_b], m)$;
$b' \leftarrow \mathcal{A}^{\mathbf{GSig}(\cdot,\cdot)}(\sigma)$;
return $b'$ if :
    $\mathcal{CT} \bigcap \{i_0, i_1\} = \emptyset, \mathcal{CK} \bigcap \{i_0, i_1\} = \emptyset$;
else return 0.
```

Figure C2 Insider-anonymity for VLR-GS.

Traceability. As shown in Fig. C3, the adversary aims to produce a valid message-signature pair with some revocation list, such that the specified tracing algorithm either fails or outputs someone not in the adversarial coalition. As a special case, traceability implies the basic requirement of *unforgeability* [5] for digital signature when $\mathcal{CK} = \emptyset$.

Definition 9. A VLR-GS scheme is traceable, if for any *p.p.t.* adversary $\mathcal{A}$, the following advantage function is negligible:

$$adv^{tr}_{\mathcal{A},\mathcal{GS}}(n) = Pr[1 \leftarrow \mathbf{Expt}^{tr}_{\mathcal{A},\mathcal{GS}}(1^n, N)].$$

```
$\mathbf{Expt}^{tr}_{\mathcal{A},\mathcal{GS}}(1^n, N)$
$(gpk, grt, gsk) \leftarrow \mathsf{GKg}(1^n, 1^N)$;
$(m, \sigma, RL) \leftarrow \mathcal{A}^{\mathbf{GSig}(\cdot,\cdot),\mathbf{CorruptK}(\cdot)}(gpk, grt)$;
return $1$ if :
    $\mathsf{GVf}(gpk, RL, m, \sigma) = 1$,
    $(m, \sigma)$ traces to someone out of $\mathcal{CK} \backslash RL$ or the tracing fails,
    $\mathbf{GSig}(m, i)$ was never queried for $i \notin \mathcal{CK}$;
else return $0$.
```

Figure C3 Traceability for VLR-GS.

Appendix D The model of AE

Same as regular PKE, an AE scheme consists of three *p.p.t.* algorithms:

1. $\mathsf{Kg}(1^n)$: On input the security parameter, the key-generation algorithm outputs a pair of public/secret keys (pk, sk).
2. $\mathsf{Enc}(pk, m)$: The encryption algorithm takes as input the public key as well as some plaintext, and it outputs a ciphertext c.
3. $\mathsf{Dec}(sk, c)$: The decryption algorithm takes as input the secret key and some ciphertext, and it outputs a plaintext m or a symbol $\perp$ indicating a decryption failure.

Definition 10. An AE scheme is correct, if for any $n \in \mathbb{N}$, $(pk, sk) \in [\mathsf{Kg}(1^n)]$, and any plaintext m, it holds with overwhelming probability that:

$$\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m.$$

```
$\mathbf{Expt}^{ind,b}_{\mathcal{A},\mathcal{AE}}(1^n)$
$(pk, sk) \leftarrow \mathsf{Kg}(1^n)$;
$(m_0, m_1) \leftarrow \mathcal{A}(pk)$;
$c \leftarrow \mathsf{Enc}(pk, m_b)$;
$b' \leftarrow \mathcal{A}(c)$;
return $b'$.
```

Figure D1 Indistinguishability for AE.

```
$\mathbf{Expt}^{kp,b}_{\mathcal{A},\mathcal{AE}}(1^n)$
$(pk_0, sk_0) \leftarrow \mathsf{Kg}(1^n), (pk_1, sk_1) \leftarrow \mathsf{Kg}(1^n)$;
$m \leftarrow \mathcal{A}(pk_0, pk_1)$;
$c \leftarrow \mathsf{Enc}(pk_b, m)$;
$b' \leftarrow \mathcal{A}(c)$;
return $b'$.
```

Figure D2 Key-privacy for AE.

Security Properties. Besides the basic requirement of *indistinguishability* [6] for regular PKE (see Fig. D1), AE additionally holds *key-privacy* [7], which is formalized by the indistinguishability experiment with roles of public keys and plaintexts reversed (see Fig. D2). Intuitively, the ciphertext generated by AE not only conceals the original plaintext, but also hides the public key under which it was encrypted.

Definition 11. An AE scheme is called secure, if both the following advantage functions are negligible for any *p.p.t.* adversary $\mathcal{A}$:

$$adv^{ind}_{\mathcal{A},\mathcal{AE}}(n) = |Pr[1 \leftarrow \mathbf{Expt}^{ind,0}_{\mathcal{A},\mathcal{AE}}(1^n)] - Pr[1 \leftarrow \mathbf{Expt}^{ind,1}_{\mathcal{A},\mathcal{AE}}(1^n)]|;$$
$$adv^{kp}_{\mathcal{A},\mathcal{AE}}(n) = |Pr[1 \leftarrow \mathbf{Expt}^{kp,0}_{\mathcal{A},\mathcal{AE}}(1^n)] - Pr[1 \leftarrow \mathbf{Expt}^{kp,1}_{\mathcal{A},\mathcal{AE}}(1^n)]|.$$

Appendix E The (refined) model of VLR-HGS

We refer readers to Table E1 for symbols used in depicting VLR-HGS. In syntax, a VLR-HGS scheme consists of three *p.p.t.* algorithms:

1. $\mathsf{HKg}(1^n, \mathcal{T})$: On input the security parameter and some tree depicting the group structure, this algorithm outputs a tuple of maps (hpk, hsk, hrt). The former two maps associate each $\alpha \in \mathcal{T}$ with a public value $hpk(\alpha)$ and a secret value $hsk(\alpha)$ respectively, while the map hrt specifies the revocation token $hrt(\alpha)$ for signer $\alpha \in \mathcal{L}(\mathcal{T})$.
2. $\mathsf{HSig}(hpk, hsk(\alpha), m)$: The signing algorithm takes as input the public map, the secret key of some signer and a message; it outputs a signature σ.
3. $\mathsf{HVf}(hpk, RL, m, \sigma)$: The verification algorithm takes as input a public map, a public revocation list $RL \subseteq \{hrt(\alpha)\}_{\alpha \in \mathcal{L}(\mathcal{T})}$ and a candidate message-signature pair; it outputs a bit b, and if $b = 0$, it means either (m, σ) is invalid or its signer has been revoked.

Table E1 Symbols used in depicting VLR-HGS.

Symbol	Description
$\mathcal{T}$	a balanced tree with size polynomial in n.
δ	the depth of $\mathcal{T}$.
ρ	the root of $\mathcal{T}$.
$\mathcal{L}(\mathcal{T})$	all signer/leaf nodes in $\mathcal{T}$.
S_α	signer/leaf node with index α.
M_β	the manager/inner node with β as indexes of all its direct children.
$child(\beta)$	the index set of all direct children of manager β.
$\mathcal{T}^i$	all nodes at depth $i \in \{0, 1, \cdots, \delta\}$; specifically, $\mathcal{T}^\delta = \mathcal{L}(\mathcal{T})$.
$\mathsf{Co}_{\{\alpha^{(0)},\alpha^{(1)}\}}$	all nodes from some two leaves $\alpha^{(0)}, \alpha^{(1)} \in \mathcal{L}(\mathcal{T})$ to their first common ancestor, with $\alpha^{(0)}, \alpha^{(1)}$ excluded.

Remark 1. Our description is slightly different from the original syntax [8]. Specifically, now the domain of hrt consists of only signer nodes. In fact, either VLR-GS or VLR-HGS is devoted to solve the issue of revoking the signing capability *rather than* the tracing capability, thus it is syntactically unnecessary to generate tokens for managers. Hou et al. [8] chose to do so because those tokens would be inputs to their tracing algorithm. However, such purpose is not essential to our construction, since more efficient tracing strategies are employed.

On finding more efficient tracing algorithm. For general discussions, we denote the tracing algorithm explicitly by HOpen, and separate the original definition for a correct HGS [8] into two aspects, namely correct verification (Def. 12) and correct tracing (Def. 13). In [8], the verification algorithm is employed to do the tracing, thus the correctness of the latter is inherited from that of the former (in their syntax). However, an efficiency concern is inherent to their method, since the tracing cost of a manager is linear to its children number. Similar issue has been well solved in the context of VLR-GS, e.g., in a generic sense [9], and our construction presents a more efficient tracing for VLR-HGS.

Definition 12. Given a VLR-HGS scheme, we say its verification algorithm is correct, if for any n, $\mathcal{T}$, $\alpha \in \mathcal{L}(\mathcal{T})$, m and $(hpk, hsk, hrt) \in [\mathsf{HKg}(1^n, \mathcal{T})]$, it holds except with negligible probability that:

$$\mathsf{HVf}(hpk, RL, m, \mathsf{HSig}(hpk, hsk(\alpha), m)) = 1 \iff hrt(\alpha) \notin RL.$$

Definition 13. Given a VLR-HGS scheme, its tracing algorithm HOpen is correct, if for any n, $\mathcal{T}$, $\alpha \in \mathcal{L}(\mathcal{T})$, m and $(hpk, hsk, hrt) \in [\mathsf{HKg}(1^n, \mathcal{T})]$, let $\sigma = \mathsf{HSig}(hpk, hsk(\alpha), m)$ and let $\beta_0 := \rho \ni \beta_1 \ni \cdots \ni \beta_\delta := \alpha$ be the path from the root to the signer, and it holds with overwhelming probability that:

$$\mathsf{HOpen}(hpk, hsk(\beta_i), m, \sigma) = \beta_{i+1}, \text{ for all } i = 0, 1, \cdots, \delta - 1.$$

Anonymity. This property preserves the identity privacy of honest signers. *Full-anonymity* [8] for VLR-HGS is depicted in Fig. E1, where the adversary is given full signing keys. In this paper, we additionally define *insider-anonymity* for VLR-HGS (Fig. E2), which by contrast denies $\mathcal{A}$ the signing keys of two challenge identities. Comparisons between these two versions of anonymity are similar to what we have argued in the context of VLR-GS. The following oracles are used in the experiments:

1. $\mathbf{HSig}(\cdot,\cdot)$: on queries $(m, \alpha \in \mathcal{L}(\mathcal{T}))$, return $\sigma \leftarrow \mathsf{HSig}(hpk, hsk(\alpha), m)$;
2. $\mathbf{HCorruptK}(\cdot)$: a set $\mathcal{HCK}$ is initialized as $\emptyset$; on queries $\alpha \in \mathcal{T}$, return the secret key $hsk(\alpha)$, and add α into $\mathcal{HCK}$;
3. $\mathbf{HCorruptT}(\cdot)$: a set $\mathcal{HCT}$ is initialized as $\emptyset$; on queries $\alpha \in \mathcal{L}(\mathcal{T})$, return the revocation token $hrt(\alpha)$, and add α into $\mathcal{HCT}$.

To exclude trivial cases, $hrt(\alpha^{(0)})$ and $hrt(\alpha^{(1)})$ should never be given to $\mathcal{A}$ assuming the correctness of verification; besides, $\mathcal{A}$ cannot obtain $hsk(\alpha)$ with any $\alpha \in \mathsf{Co}_{\{\alpha^{(0)},\alpha^{(1)}\}}$ if the tracing algorithm is correct. Note that the adversary has formal access to $\mathbf{HCorruptK}(\cdot)$ and $\mathbf{HCorruptT}(\cdot)$ only in challenge phase, because once the challenge identities are decided on, unrestricted secret keys and revocation tokens can be obtained at once by the adversary.

Definition 14. A VLR-HGS scheme is fully anonymous, if for any *p.p.t.* adversary $\mathcal{A}$, the following advantage function is negligible:

$$adv^{fa}_{\mathcal{A},\mathcal{HGS}}(n) = |Pr[1 \leftarrow \mathbf{Expt}^{fa,0}_{\mathcal{A},\mathcal{HGS}}(1^n, \mathcal{T})] - Pr[1 \leftarrow \mathbf{Expt}^{fa,1}_{\mathcal{A},\mathcal{HGS}}(1^n, \mathcal{T})]|.$$

Definition 15. A VLR-HGS scheme is insider-anonymous, if for any *p.p.t.* adversary $\mathcal{A}$, the following advantage function is negligible:

$$adv^{ia}_{\mathcal{A},\mathcal{HGS}}(n) = |Pr[1 \leftarrow \mathbf{Expt}^{ia,0}_{\mathcal{A},\mathcal{HGS}}(1^n, \mathcal{T})] - Pr[1 \leftarrow \mathbf{Expt}^{ia,1}_{\mathcal{A},\mathcal{HGS}}(1^n, \mathcal{T})]|.$$

```
$\mathbf{Expt}^{fa,b}_{\mathcal{A},\mathcal{HGS}}(1^n,\mathcal{T})$
$(hpk, hsk, hrt) \leftarrow \mathsf{HKg}(1^n, \mathcal{T});$
$(\alpha^{(0)}, \alpha^{(1)}, m) \leftarrow \mathcal{A}^{\mathbf{HCorruptT}(\cdot),\mathbf{HCorruptK}(\cdot)}(\{hsk(\alpha)\}_{\alpha\in\mathcal{L}(\mathcal{T})}, hpk);$
$\sigma \leftarrow \mathsf{HSig}(hpk, hsk(\alpha^{(b)}), m);$
$b' \leftarrow \mathcal{A}(\sigma);$
$\textbf{return } b' \textbf{ if }:$
    $\mathcal{HCT}\bigcap\{\alpha^{(0)}, \alpha^{(1)}\} = \emptyset,$
    $\mathcal{HCK}\bigcap\mathsf{Co}_{\{\alpha^{(0)},\alpha^{(1)}\}} = \emptyset;$
$\textbf{else return } 0.$
```

Figure E1 Full-anonymity for VLR-HGS.

```
$\mathbf{Expt}^{ia,b}_{\mathcal{A},\mathcal{HGS}}(1^n,\mathcal{T})$
$(hpk, hsk, hrt) \leftarrow \mathsf{HKg}(1^n, \mathcal{T});$
$(\alpha^{(0)}, \alpha^{(1)}, m) \leftarrow \mathcal{A}^{\mathbf{HCorruptT}(\cdot),\mathbf{HCorruptK}(\cdot),\mathbf{HSig}(\cdot,\cdot)}(hpk);$
$\sigma \leftarrow \mathsf{HSig}(hpk, hsk(\alpha^{(b)}), m);$
$b' \leftarrow \mathcal{A}^{\mathbf{HSig}(\cdot,\cdot)}(\sigma);$
$\textbf{return } b' \textbf{ if }:$
    $(\mathcal{HCT}\bigcup\mathcal{HCK})\bigcap\{\alpha^{(0)}, \alpha^{(1)}\} = \emptyset,$
    $\mathcal{HCK}\bigcap\mathsf{Co}_{\{\alpha^{(0)},\alpha^{(1)}\}} = \emptyset;$
$\textbf{else return } 0.$
```

Figure E2 Insider-anonymity for VLR-HGS.

$\mathbf{Expt}^{tr}_{\mathcal{A},\mathcal{HGS}}(1^n,\mathcal{T})$

```
$(hpk, hsk, hrt) \leftarrow \mathsf{HKg}(1^n, \mathcal{T})$;
$(m, \sigma, RL) \leftarrow \mathcal{A}^{\mathbf{HSig}(\cdot,\cdot),\mathbf{HCorruptK}(\cdot)}(hpk, \{hsk(\beta)\}_{\beta \in (\mathcal{T}-\mathcal{L}(\mathcal{T}))}, hrt)$;
return 1 if :
    $\mathsf{HVf}(hpk, RL, m, \sigma) = 1$,
    $[\mathsf{HOpen}(hpk, hsk(\mathcal{T}^{\delta-1}), m, \sigma)] = \{\perp\}$, or
    $[\mathsf{HOpen}(hpk, hsk(\mathcal{T}^{\delta-1}), m, \sigma)] \bigcap (\mathcal{L}(\mathcal{T}) - \mathcal{HCK} \backslash RL) \neq \emptyset$,
    $\mathbf{HSig}(m, \alpha)$ was never queried for $\alpha \notin \mathcal{HCK}$;
else return 0.
```

Figure E3 Traceability for VLR-HGS.

Traceability. This property depicts the robustness of a VLR-HGS system in terms of tracing. Given full revocation tokens and the secret keys of all managers, the adversary can corrupt signers by querying $\mathbf{HCorruptK}(\cdot)$, and obtain signatures of the honest by querying $\mathbf{HSig}(\cdot,\cdot)$. A success will be claimed, if $\mathcal{A}$ manages to output a valid message-signature pair (m,σ) with some revocation list RL, such that all managers at the penultimate depth fail to trace, or there exists a manager who opens this to someone out of the adversarial coalition.

For a traceable VLR-HGS scheme with correct tracing algorithm, a valid message-signature pair *cannot* be (with overwhelming probability) honestly generated if the hierarchical tracing fails. In that case, misbehaviors will always be found out by a joint work of all managers at the penultimate depth (*whole-depth tracing* in [8]) and nobody will be framed. Such mechanisms reflect a *detect-then-punish* paradigm [10], and in practice, the whole-depth tracing would merely happen considering the harsh punishments of misbehaving. On the other hand, if the hierarchical tracing does not fail, it will always locate the actual generator not revoked. Besides the security it offers, such traceability definition is quite attractive to us since it facilitates our construction as we will see.

Definition 16. An VLR-HGS scheme is traceable, if for any *p.p.t.* adversary $\mathcal{A}$, the following advantage function is negligible:

$$adv^{tr}_{\mathcal{A},\mathcal{HGS}}(n) = Pr[1 \leftarrow \mathbf{Expt}^{tr}_{\mathcal{A},\mathcal{HGS}}(1^n,\mathcal{T})].$$

Appendix F Proof of verification correctness

Proof. Let $(m, \sigma := (c_0, c_1, \cdots, c_{\delta-2}, \sigma'))$ be a message-signature pair honestly generated by some signer α. By construction, $\mathsf{HVf}(hpk, RL, m, \sigma) = 1$ iff. $\mathsf{GVf}(gpk, RL, m, \sigma') = 1$. From the correctness of VLR-GS, $\mathsf{GVf}(gpk, RL, m, \sigma') = 1 \iff grt[\alpha] \notin RL$ with overwhelming probability. Thus, $\mathsf{HVf}(hpk, RL, m, \sigma) = 1 \iff hrt(\alpha) \notin RL$ with overwhelming probability.

Appendix G Proof of tracing correctness

Proof. Let $(m, \sigma := (c_0, c_1, \cdots, c_{\delta-2}, \sigma'))$ be a message-signature pair honestly generated by some signer α, and let $\beta_0 := \rho \ni \beta_1 \ni \cdots \ni \beta_\delta := \alpha$ denote the path from the root to the signer. For managers β_i, $i = 0, 1, \cdots, \delta-2$, it holds with overwhelming probability that $\mathsf{Dec}(hsk(\beta_i), c_i) = \beta_{i+1}$, assuming the correctness of AE; for the manager $\beta_{\delta-1}$, first, $grt[\alpha] \in hsk(\beta_{\delta-1})$ by our construction, and it holds with overwhelming probability that $\mathsf{GVf}(gpk, \{grt[\alpha']\}, m, \sigma') = 0 \iff grt[\alpha'] = grt[\alpha] \iff \alpha' = \alpha$ by the correctness of VLR-GS. Taking both into consideration, it follows easily the correctness of our tracing algorithm since the tree depth δ is polynomially bounded in n.

Appendix H Proof of anonymity

Proof. Let $\mathcal{A}$ denote a *p.p.t.* adversary against our construction in terms of insider-anonymity. Let $\alpha^{(0)} \in \alpha_1^{(0)} \in \cdots \in \alpha_t^{(0)} = \alpha_t^{(1)} \ni \alpha_{t-1}^{(1)} \ni \cdots \ni \alpha^{(1)}$ be nodes from two challenge identities $\alpha^{(0)}, \alpha^{(1)}$ to their first common ancestor $\alpha_t^{(0)} = \alpha_t^{(1)}$, $1 \leqslant t \leqslant \delta$. We consider a sequence of experiments defined as follows:

1. $\mathbf{Expt}^0$: defined exactly by adapting our construction to the experiment $\mathbf{Expt}^{ia,0}_{\mathcal{A},\mathcal{HGS}}(1^n,\mathcal{T})$ as shown in **Fig.** E2;
2. $\mathbf{Expt}^1$: the same as $\mathbf{Expt}^0$ except that the $(\delta-1)^{st}$ component in the outputted signature $\sigma := (c_0, c_1, \cdots, c_{\delta-2}, \sigma')$ is generated by: $c_{\delta-2} \leftarrow \mathsf{Enc}(pk_{\alpha_1^{(1)}}, \alpha^{(1)})$;
   ...

$i+1$. $\mathbf{Expt}^i$: the same as $\mathbf{Expt}^{i-1}$ except that the $(\delta-i)^{th}$ component in the signature is generated by: $c_{\delta-i-1} \leftarrow \mathsf{Enc}(pk_{\alpha_i^{(1)}}, \alpha_{i-1}^{(1)})$;
...

$t+1$. $\mathbf{Expt}^t$: the same as $\mathbf{Expt}^{t-1}$ except that the $(\delta - t)^{th}$ component in the signature is generated by: $c_{\delta-t-1} \leftarrow \mathsf{Enc}(pk_{\alpha_t^{(1)}}, \alpha_{t-1}^{(1)})$;

$t+2$. $\mathbf{Expt}^{t+1}$: defined exactly by adapting our construction to the experiment $\mathbf{Expt}^{ia,1}_{\mathcal{A},\mathcal{HGS}}(1^n, \mathcal{T})$ as shown in **Fig.** E2.

By **Def.** 15 and the Triangular Inequality, we have:

$$\begin{aligned} adv^{ia}_{\mathcal{A},\mathcal{HGS}}(n) =& |Pr[\mathbf{Expt}^0 = 1] - Pr[\mathbf{Expt}^{t+1} = 1]| \\ =& |Pr[\mathbf{Expt}^0 = 1] - Pr[\mathbf{Expt}^1 = 1] + Pr[\mathbf{Expt}^1 = 1] - Pr[\mathbf{Expt}^2 = 1] \\ &+ \cdots + Pr[\mathbf{Expt}^t = 1] - Pr[\mathbf{Expt}^{t+1} = 1]| \\ \leqslant& |Pr[\mathbf{Expt}^0 = 1] - Pr[\mathbf{Expt}^1 = 1]| + |Pr[\mathbf{Expt}^1 = 1] - Pr[\mathbf{Expt}^2 = 1]| \\ &+ \cdots + |Pr[\mathbf{Expt}^t = 1] - Pr[\mathbf{Expt}^{t+1} = 1]|. \end{aligned}$$

Claim 1: For $i = 0, 1, \cdots, t-1$, $|Pr[\mathbf{Expt}^i = 1] - Pr[\mathbf{Expt}^{i+1} = 1]| \leqslant \mathsf{negl}_i(n)$.

Except for $c_{\delta-i-2} \leftarrow \mathsf{Enc}(pk_{\alpha_{i+1}^{(0)}}, \alpha_i^{(0)})$ in $\mathbf{Expt}^i$ and $c_{\delta-i-2} \leftarrow \mathsf{Enc}(pk_{\alpha_{i+1}^{(1)}}, \alpha_i^{(1)})$ in $\mathbf{Expt}^{i+1}$, the distributions of $\sigma := (c_0, c_1, \cdots, c_{\delta-2}, \sigma')$ are identical. Since $\mathcal{A}$ cannot obtain $hsk(\alpha)$ for $\alpha \in \mathsf{Co}_{\{\alpha^{(0)}, \alpha^{(1)}\}}$, $\mathcal{A}$ has neither $sk_{\alpha_{i+1}^{(0)}}$ nor $sk_{\alpha_{i+1}^{(1)}}$. From the indistinguishability and key privacy of AE, we have:

$$\begin{aligned} &|Pr[\mathcal{A}(c_{\delta-i-2}) = 1 : c_{\delta-i-2} \leftarrow \mathsf{Enc}(pk_{\alpha_{i+1}^{(0)}}, \alpha_i^{(0)})] \\ &\quad - Pr[\mathcal{A}(c_{\delta-i-2}) = 1 : c_{\delta-i-2} \leftarrow \mathsf{Enc}(pk_{\alpha_{i+1}^{(0)}}, \alpha_i^{(1)})]| = \mathsf{negl}_i^{(1)}(n). \\ &|Pr[\mathcal{A}(c_{\delta-i-2}) = 1 : c_{\delta-i-2} \leftarrow \mathsf{Enc}(pk_{\alpha_{i+1}^{(0)}}, \alpha_i^{(1)})] \\ &\quad - Pr[\mathcal{A}(c_{\delta-i-2}) = 1 : c_{\delta-i-2} \leftarrow \mathsf{Enc}(pk_{\alpha_{i+1}^{(1)}}, \alpha_i^{(1)})]| = \mathsf{negl}_i^{(2)}(n). \end{aligned}$$

Considering that each component is independently generated, we have:

$$\begin{aligned} &|Pr[\mathbf{Expt}^i = 1] - Pr[\mathbf{Expt}^{i+1} = 1]| \\ =& |Pr[\mathcal{A}(c_{\delta-i-2}) = 1 : c_{\delta-i-2} \leftarrow \mathsf{Enc}(pk_{\alpha_{i+1}^{(0)}}, \alpha_i^{(0)})] \\ &- Pr[\mathcal{A}(c_{\delta-i-2}) = 1 : c_{\delta-i-2} \leftarrow \mathsf{Enc}(pk_{\alpha_{i+1}^{(1)}}, \alpha_i^{(1)})]| \\ \leqslant& |Pr[\mathcal{A}(c_{\delta-i-2}) = 1 : c_{\delta-i-2} \leftarrow \mathsf{Enc}(pk_{\alpha_{i+1}^{(0)}}, \alpha_i^{(0)})] \\ &- Pr[\mathcal{A}(c_{\delta-i-2}) = 1 : c_{\delta-i-2} \leftarrow \mathsf{Enc}(pk_{\alpha_{i+1}^{(0)}}, \alpha_i^{(1)})]| \\ &+ |Pr[\mathcal{A}(c_{\delta-i-2}) = 1 : c_{\delta-i-2} \leftarrow \mathsf{Enc}(pk_{\alpha_{i+1}^{(0)}}, \alpha_i^{(1)})] \\ &- Pr[\mathcal{A}(c_{\delta-i-2}) = 1 : c_{\delta-i-2} \leftarrow \mathsf{Enc}(pk_{\alpha_{i+1}^{(1)}}, \alpha_i^{(1)})]| \\ =& \mathsf{negl}_i(n) := \mathsf{negl}_i^{(1)}(n) + \mathsf{negl}_i^{(2)}(n). \end{aligned}$$

Claim 2: $|Pr[\mathbf{Expt}^t = 1] - Pr[\mathbf{Expt}^{t+1} = 1]| \leqslant \mathsf{negl}_t(n)$.

Similarly, two distributions of $\sigma := (c_0, c_1, \cdots, c_{\delta-2}, \sigma')$ are identical, except that $\sigma' \leftarrow \mathsf{GSig}(gpk, hsk(\alpha^{(0)}), m)$ in $\mathbf{Expt}^t$ and $\sigma' \leftarrow \mathsf{GSig}(gpk, hsk(\alpha^{(1)}), m)$ in $\mathbf{Expt}^{t+1}$. As requested, $\mathcal{A}$ cannot obtain $hsk(\alpha) = gsk[\alpha]$ or $hrt(\alpha) = grt[\alpha]$ with $\alpha \in \{\alpha^{(0)}, \alpha^{(1)}\}$. The oracle **HSig** can be perfectly simulated by the oracle **GSig** accompanied with public key encryption, then by the insider anonymity of VLR-GS and the independence of all components, we have:

$$\begin{aligned} &|Pr[\mathbf{Expt}^t = 1] - Pr[\mathbf{Expt}^{t+1} = 1]| \\ =& |Pr[\mathcal{A}(\sigma') = 1 : \sigma' \leftarrow \mathsf{GSig}(gpk, hsk(\alpha^{(0)}), m)] \\ &- Pr[\mathcal{A}(\sigma') = 1 : \sigma' \leftarrow \mathsf{GSig}(gpk, hsk(\alpha^{(1)}), m)]| = \mathsf{negl}_t(n). \end{aligned}$$

Finally, we have:

$$adv^{ia}_{\mathcal{A},\mathcal{HGS}}(n) \leqslant \mathsf{negl}(n) := \sum_{i=0}^{t} \mathsf{negl}_i(n).$$

Another sequence of experiments $(\mathbf{Expt'}^0, \mathbf{Expt'}^1, \cdots, \mathbf{Expt'}^{t+1})$ can be similarly defined to prove full-anonymity, by replacing the experiment $\mathbf{Expt}^{ia,b}_{\mathcal{A},\mathcal{HGS}}(1^n, \mathcal{T})$ with $\mathbf{Expt}^{fa,b}_{\mathcal{A},\mathcal{HGS}}(1^n, \mathcal{T})$ as shown in **Fig.** E1. By exactly the same proof, $|Pr[\mathbf{Expt'}^i = 1] - Pr[\mathbf{Expt'}^{i+1} = 1]| \leqslant \mathsf{negl}'_i(n)$ for $i = 0, 1, \cdots, t-1$, and by essentially the same proof without discussions on the oracle **HSig**, $|Pr[\mathbf{Expt'}^t = 1] - Pr[\mathbf{Expt'}^{t+1} = 1]| \leqslant \mathsf{negl}'_t(n)$. Therefore,

$$adv^{fa}_{\mathcal{A},\mathcal{HGS}}(n) \leqslant \mathsf{negl}'(n) := \sum_{i=0}^{t} \mathsf{negl}'_i(n).$$

Appendix I Proof of traceability

Proof. Let $\mathcal{A}$ be a *p.p.t.* adversary against our construction in terms of traceability. For contradiction, assume $\mathcal{A}$ succeeds in $\mathbf{Expt}^{tr}_{\mathcal{A},\mathcal{HGS}}(1^n, \mathcal{T})$ in **Fig.** E3, namely $\mathcal{A}$ manages to output an $(m, \sigma := (c_0, c_1, \cdots, c_{\delta-2}, \sigma'), RL)$ such that:

1. $\mathsf{GVf}(gpk, RL, m, \sigma') = 1$;
2. $\mathsf{GVf}(gpk, \{grt[\alpha]\}, m, \sigma') = 1$ for all $\alpha \in \mathcal{L}(\mathcal{T})$, or there exists some $\alpha \notin (\mathcal{HCK} \backslash RL)$ such that $\mathsf{GVf}(gpk, \{grt[\alpha]\}, m, \sigma') = 0$;
3. $\mathbf{HSig}(m, \alpha)$ was never queried for $\alpha \notin \mathcal{HCK}$.

Besides, the oracle **HSig** can be perfectly simulated by the oracle **GSig** accompanied with some public key encryptions. Thus, $\mathcal{A}$ will also succeed in the traceability experiment for VLR-GS in Fig. C3. Then we have:

$$Pr[1 \leftarrow \mathbf{Expt}^{tr}_{\mathcal{A},\mathcal{HGS}}(1^n, \mathcal{T})] \leqslant Pr[1 \leftarrow \mathbf{Expt}^{tr}_{\mathcal{A},\mathcal{GS}}(1^n, |\mathcal{L}(\mathcal{T})|)] = \mathsf{negl}(n).$$

References

1 Micciancio D, Regev O. Worst-case to average-case reductions based on gaussian measures. SIAM J. Comput., 2007, **37**(1): 267–302

2 Regev O. On lattices, learning with errors, random linear codes, and cryptography. In: Gabow H N, Fagin R eds. Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, 2005. 84–93

3 Gentry C, Peikert C, Vaikuntanathan V. Trapdoors for hard lattices and new cryptographic constructions. In: Dwork C eds. Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, 2018. 197–206

4 Boneh D, Shacham H. Group signatures with verifier-local revocation. In: Atluri V, Pfitzmann B, McDaniel P D eds. Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, 2004. 168–177

5 Goldwasser S, Micali S, Rivest R L. A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput, 1988, **17**(2): 281–308

6 Goldwasser S, Micali S. Probabilistic encryption. J. Comput. Syst. Sci, 1984, **28**(2): 270–299

7 Bellare M, Boldyreva A, Desai A, et al. Key-privacy in public-key encryption. In: Boyd C eds. Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, 2001. 566–582

8 Hou L, Liu R, Qiu T, et al. Hierarchical group signatures with verifier-local revocation. In: Naccache D, Xu S, Qing S, et al. eds. Information and Communications Security - 20th International Conference, ICICS, Lille, France, 2018. 271–286

9 Ishida A, Sakai Y, Emura K, et al. Fully anonymous group signature with verifier-local revocation. In: Catalano D, Prisco R D eds. Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, 2018. 23–42

10 Wayner P. Digital Cash: Commerce on the net. 2ed edn. Academic Press, Cambridge. 1997