Sun and Liu *Cybersecurity* (2021) 4:15
https://doi.org/10.1186/s42400-021-00076-8

Cybersecurity

RESEARCH Open Access

An efficient fully dynamic group signature with message dependent opening from lattice

Yiru Sun[1,2,3*] and Yanyan Liu[1,2,3]

Abstract

Message-dependent opening is one of the solutions to solve the problem of the tracing manager owns excessive power. In this paper, we present a new lattice-based fully dynamic group signature scheme with message-dependent opening by combining an improved version of the fully dynamic group signature scheme proposed by Ling et al and the double encryption paradigm. In addition, we propose an improved underlying zero knowledge protocol, it has a soundness error $\frac{1}{\max(n,p)+1}$ that is better than the Stern-like protocol, which helps to bring down the communication complexity of the protocol and hence the signature scheme. Our scheme constrains the power of group managers by adding an admitter, and the signature size has a logarithmic relationship with the group size.

Keywords: Dynamic group signature, Message-dependent opening, NIZK, LWE, SIS

Introduction

Related work

Since the concept of group signature was proposed in Chaum and van Heyst (1991), it has become an important primitive to realize anonymous authentication. Group signature allows members in a group to sign messages on behalf of the group without revealing any information of the signer's identity. At the same time, the signature could be traced to the signer when it is in dispute. In other words, there is an authority in the scheme called trace manager GM_{trace} who can de-anonymize the signature and trace it to the specific signer. But in many scenarios, GM_{trace} is given too much power as it can open all signatures whether the signer is valid or not. To solve this problem, there is an extension of the group signature in Sakai et al. (2012) to balance the traceability and privacy, it is called group signature scheme with message-dependent opening (GS-MDO). In the GS-MDO system, there is another participant named admitter, and the trace manager GM_{trace} could open one signature only when he work with the admitter. To open a signature Σ of message M, the admitter generates a token $\mathbf{t_M}$ with respect to M using its secret key firstly, and sends $\mathbf{t_M}$ to the trace manager GM_{trace}, then GM_{trace} uses its secret key and $\mathbf{t_M}$ to open the signature. That is, the trace manager GM_{trace} can only open the signatures of messages specified by admitter. Subsequently, many other GS-MDO schemes were proposed based on different assumptions, such as decision linear (DLIN) (Sakai et al. 2012), strong Diffie-Hellman (Ohara et al. 2013), Decision 3-party Diffie-Hellman (D3DH) (Libert and Joye 2014), learning with error (LWE) and small integer solution (SIS) (Libert et al. 2016).

Lattice-based cryptography has attracted a lot of attention for its simple arithmetic operations and potential ability to resist quantum attack. However, compared with other non-lattice based cryptographic schemes, such as DDH, factoring, et al, the efficiency of lattice-based cryptographic schemes have not been solved well. The

*Correspondence: sunyiru@iie.ac.cn
[1]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Haidian District, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Huairou District, China
Full list of author information is available at the end of the article

first lattice-based static group signature scheme is given in Gordon et al. (2010), its security is proven in RO model, and there is a linear relationship between signature size and group size N. Subsequently, the signature size was lowered up to $O(\log N)$ by different manners (Laguillaumie et al. 2013; Ling et al. 2015), such as bonsai tree (Langlois et al. 2014), Merkle hash tree (Libert et al. 2016) and lattice-based accumulators (Ling et al. 2017). In order to further satisfy the requirements of real applications, it is possible to realize the dynamic registration and revocation of users efficiently (Ling et al. 2017) by combining the static group signature scheme in Libert et al. (2016) with the security model in Bootle et al. (2016). It includes an update algorithm in accumulator that is constructed based on hash Merkle tree, and both the security and the signature size were improved compared with the scheme in Libert et al. (2016). However, the schemes above all follow encryption-then-proof pattern, and rely heavily on zero-knowledge protocol in the proof process, which limits the improvement of efficiency and security. In order to break this bottleneck, there are currently two research lines: one is to try to remove the zero-knowledge proof protocol from the construction of group signature schemes, which is the research content in Katsumata and Yamada (2019). In other words, a lattice-based static group signature scheme without NIZK was proposed in Katsumata and Yamada (2019), and it is proved secure under the standard model. There is a natural idea: whether it is possible to construct a lattice-based fully dynamic group signature scheme that is provably secure under the standard model? To solve this problem, we tried to propose a construction in Sun and Liu (2020) and proved it to be secure under the standard model. The other is to improve the efficiency of zero-knowledge proof (Beullens 2020) and try to apply it to the construction of group signature schemes under the RO model. Our work in this paper gives a positive solution of the latter.

Our contribution

In this paper, we give a new fully dynamic group signature scheme over ring with message-dependent opening (FDGS-MDO) by combining an improved version of the fully dynamic group signature scheme in Ling et al. (2017) and the double encryption paradigm (Canetti et al. 2004), which uses our following zero knowledge proof of knowledge as a underlying protocol. Compared with the scheme in Sun et al. (2019), our scheme realizes the weaken of GM_{trace}'s power by adding another participant: admitter. Concretely, the admitter could generate tokens with respect to messages by using its secret key such that the trace manager can only open signatures of messages specified by the admitter. And we also give an improved zero knowledge proof of knowledge that has smaller soundness error than Stern-like protocol, and we use it as the underlying protocol to improves the efficiency of the scheme in Sun et al. (2019).

We give the specific construction and security analysis of our zero knowledge proof of knowledge, which partially realizes the optimization idea in Beullens (2020). In Beullens (2020), it is necessary to transform an instance of SIS problem into an instance of the permuted kernel problem (PKP) firstly, and then prove its knowledge by using a Σ - protocol for latter, while in our work, we omit this transformation operation. In addition, in order to reduce the communication complexity of our underlying protocol, the prover does not need to send all commitments $\{\mathbf{com}_{ic}\}_{i\in[n],c\in\mathbb{Z}_p}$ and $\{\mathbf{com}_i\}_{i\in[n]}$ to the verifier in the first round of our protocol. We build two Merkle hash trees with the commitments $\{\mathbf{com}_{ic}\}_{i\in[n],c\in\mathbb{Z}_p}$ and $\{\mathbf{com}_i\}_{i\in[n]}$ as leaves respectively, and send the roots $\mathbf{u}$ and $\hat{\mathbf{u}}$ of the two trees to the verifier. In the third round of the protocol, the prover needs to send some additional messages to the verifier: the commitments $\mathbf{com}_I$, $\mathbf{com}_{Ich}$ for challenge (I, ch) and the witnesses w_I, w_{Ich} that needed to recompute the roots. The verifier need to check that whether the roots $\mathbf{u}'$, $\hat{\mathbf{u}}'$ he recomputes are consistent with $\mathbf{u}$, $\hat{\mathbf{u}}$ received in the first round. Our protocol has a soundness error $\frac{1}{\max(n,p)+1}$, which is better than the soundness error $\frac{2}{3}$ of the Stern-like protocol. Given a security parameter λ, our protocol need to be executed $k' = \frac{\lambda}{\log(\max(n,p)+1)}$ times sequentially to realize a negligible soundness error $2^{-\lambda}$, while the Stern-like need to be performed $\Theta(\lambda)$ times sequentially. So our protocol satisfies stronger soundness and it effectively reduce the communication complexity of the protocol, thus bring to the group signature scheme the stronger security property and smaller signature size.

In the remainder of this paper, we start by reviewing some definitions, theorems used in the scheme, and the dynamic algorithm to construct the Merkle hash tree in "Preliminaries" section. In "Syntax and security of fully dynamic group signature with message dependent opening" section, we present the syntax of the fully dynamic group signature scheme with message dependent opening. And the detailed construction of the scheme and its security analysis are presented in "The lattice-based dynamic group signature scheme with message-dependent opening" section. Finally, we present the underlying zero knowledge protocol and its security analysis in "The improved zero-knowledge protocol of knowledge" section, and conclusion in "Conclusion" section.

Preliminaries

The background of lattice

In this section, we will review some notations, definitions and theorems used for analysing our main results.

Throughout this paper, set the security parameter λ, positive integer $n = O(\lambda)$, $p = O(\lambda)$, prime modules $q = \tilde{O}(n^{1.5})$, $k = n\lceil \log q \rceil$, $m = 2k$, and $R = \mathbb{Z}[x]/f(x)$, $f(x) = x^n + 1$, $R_q = R/qR$, given vectors $\mathbf{x} = (x_1, \cdots, x_m)$, $\mathbf{z} = (z_1, \cdots, z_m)$, integer t, then $\|\mathbf{x}\|_t = \left(\sum_{i=1}^{m} |x_i|^t\right)^{\frac{1}{t}}$ denotes its t-norm, $(\mathbf{x}|\mathbf{z})$ is a concatenation of the two vectors.

Definition 1 (The ring-**SVP** and ring-**SIVP**) *(Lyubashevsky et al. 2013) Given a ring R, let $\gamma \geq 1$, then the ring-$\mathbf{SVP}_\gamma$ problem is: given the ideal lattice $\mathcal{I}$ over R, find out a non-zero short vector $\mathbf{x} \in \mathcal{I}$, such that $\|\mathbf{x}\|_\infty \leq \gamma \cdot \lambda_1(\mathcal{I})$. And the ring-$\mathbf{SIVP}_\gamma$ problem could be defined similarly: find out n independent elements $(\mathbf{x}_1, \cdots, \mathbf{x}_n)$ in $\mathcal{I}$, such that $\|(\mathbf{x}_1, \cdots, \mathbf{x}_n)\|_\infty \leq \gamma \cdot \lambda_n(\mathcal{I})$.*

Definition 2 (The ring-$\mathbf{SIS}^\infty_{n,m,q,\beta}$) *(Ling et al. 2015; Peikert 2016)Choose m elements $a_j \xleftarrow{\$} R_q$ uniformly, let random vector $\mathbf{A} = (a_1, \cdots, a_m) \in R_q^m$, positive real number $\beta = poly(n)$, find out a non-zero short vector $\mathbf{z} = (z_1, \cdots, z_m) \in R_q^m$, $\|\mathbf{z}\|_\infty \leq \beta$, such that*

$$f_{\mathbf{A}}(\mathbf{z}) = \langle \mathbf{A}, \mathbf{z} \rangle = \mathbf{A}^\top \cdot \mathbf{z} = \sum_j a_j \cdot z_j = 0 \in R_q$$

Numerous studies (Lyubashevsky and Micciancio 2006; Peikert and Rosen 2006; Peikert and Rosen 2007; Lyubashevsky 2008; Lyubashevsky 2012) have shown that if $f(x)$ is irreducible polynomial with integer coefficients, $m > \frac{\log q}{\log(2\beta)}$, $\gamma = 16mn \log^2 n$, $q \geq \frac{\gamma\sqrt{n}}{4 \log n}$, then the problem ring-$\mathbf{SIS}^\infty_{n,m,q,\beta}$ is at least as difficult as the problem ring-$\mathbf{SVP}^\infty_\gamma$ over $\mathcal{I}$.

Definition 3 (The ring-**LWE** distribution) *(Peikert 2016)For secret element $s \in R_q$, $\mathcal{X}$ is the noise distribution in R_q with bound β, choose $a \xleftarrow{\$} R_q$, $e \xleftarrow{\$} \mathcal{X}$ uniformly, then $A_{s,\mathcal{X}} = (a, b = s \cdot a + e \mod q)$ is called the ring-**LWE** distribution in $R_q \times R_q$.*

Definition 4 (The decision ring-$\mathbf{LWE}_{n,m,q,\mathcal{X}}$) *(Lyubashevsky et al. 2010; Peikert 2016) Let $n, m \geq 1$, $q \geq 2$, given m samples $(a_j, b_j) \in R_q \times R_q$, which are sampled from one of the two distributions: $A_{s,\mathcal{X}}$ and the uniform distribution in $R_q \times R_q$, then the decision ring-$\mathbf{LWE}_{n,m,q,\mathcal{X}}$ is to distinguish which one the samples are from.*

Theorem 1 *(Lyubashevsky et al. 2010)Let $q = 1 \mod 2n$, $\beta \geq \omega(\sqrt{n \log n})$, $\gamma = n^2(q/\beta)(nm/\log(nm))^{1/4}$, then there is an error distribution $\mathcal{X}$ with bound β, such that the problem ring-$\mathbf{LWE}_{n,m,q,\mathcal{X}}$ is at least as difficult as the problem ring-$\mathbf{SVP}^\infty_\gamma$ over $\mathcal{I}$.*

The sigma protocol

Definition 5 (The Σ-protocol) *(Hazay and Lindell 2010)Given an NP relation $R = (x, w) \in \{0,1\}^* \times \{0,1\}^*$, a two party interactive protocol $\langle P, V \rangle$ is called Σ-protocol for relation R if it is a three-round public-coin protocol and satisfies the following requirements:*

***Completeness**: For $(x, w) \in R$, if both prover P and verifier V follow this protocol, then $\Pr[\langle P(x, w), V(x) \rangle = 1] = 1$.*

***2-Special soundness**: For any statement x, if there is an adversary $\mathcal{A}$ that outputs with noticeable probability a pair of accepting transcripts (a, e, z) and (a, e', z') with $e \neq e'$, then one can extract a witness w such that $(x, w) \in R$.*

***Special honest verifier zero knowledge**: For $(x, w) \in R$, there is a PPT simulator $\mathcal{S}$ that given the statement x and a random challenge e outputs a transcript (a,e,z) that is indistinguishable from the probability distribution of transcripts of honest executions of the protocol on input $(x, w) \in R$, i.e. $\mathcal{S}(x, e) \approx \langle P(x, w), V(x, e) \rangle$.*

The zero-knowledge protocol used in this paper satisfies completeness, $\max(n, p)+1$-special soundness and special honest-verifier zero knowledge, which depends heavily on the security (statistical hiding and computing binding) of the commitment scheme that used as a submodule in our zero-knowledge protocol. The detailed construction of our protocol and its security proof is given in "The improved zero-knowledge protocol of knowledge" section.

The dynamic algorithm of constructing lattice-based Merkle hash tree

The security of Merkle tree used in Sun et al. (2019) and here are all based on the collision-resistant hash functions, whereas the size and depth of the former are fixed, and that of the latter increase with the registration of users. For any $t \in R_q$, $\mathbf{bin}(t) \in \{0,1\}^k$ is its binary representation, let

$$\mathbf{G} = \begin{bmatrix} 1, 2, 4, \cdots, 2^{\lceil \log q \rceil - 1} & & \\ & \cdots & \\ & & 1, 2, 4, \cdots, 2^{\lceil \log q \rceil - 1} \end{bmatrix} \in \mathbb{Z}_q^{n \times k} \tag{1}$$

then $t = \mathbf{G} \cdot \mathbf{bin}(t)$. let $\mathcal{H} = \{h_{\mathbf{A}} | \mathbf{A} \xleftarrow{\$} R_q^m\}$, $h_{\mathbf{A}} : \{0,1\}^k \times \{0,1\}^k \rightarrow \{0,1\}^k$ is collision-resistant hash functions based on the ring-**SIS** problem, where $\mathbf{A} = [\mathbf{A}_0 | \mathbf{A}_1] \in R_q^m$, $\mathbf{A}_0, \mathbf{A}_1 \xleftarrow{\$} R_q^k$ is an instance of the ring-$\mathbf{SIS}_{m,q,1}$ problem, for arbitrary $(\mathbf{u}_0, \mathbf{u}_1) \in \{0,1\}^k \times \{0,1\}^k$, we have

$$h_{\mathbf{A}}(\mathbf{u}_0, \mathbf{u}_1) = \mathbf{bin}(\mathbf{A}_0 \cdot \mathbf{u}_0 + \mathbf{A}_1 \cdot \mathbf{u}_1 \mod q) \in \{0,1\}^k$$

so the following equivalent relationship is true,

$$h_{\mathbf{A}}(\mathbf{u}_0, \mathbf{u}_1) = \mathbf{u} \Leftrightarrow \mathbf{A}_0 \cdot \mathbf{u}_0 + \mathbf{A}_1 \cdot \mathbf{u}_1 = \mathbf{G} \cdot \mathbf{u} \mod q$$

Suppose that there is an PPT adversary who can give two different $\mathbf{u} \neq \mathbf{u}'$ such that $h_{\mathbf{A}}(\mathbf{u}) = h_{\mathbf{A}}(\mathbf{u}')$, then we have $\mathbf{Au} \mod q = \mathbf{Au}' \mod q$, i.e. $\mathbf{A}(\mathbf{u} - \mathbf{u}') = 0 \mod q$. Since $\mathbf{u} \neq \mathbf{u}', \mathbf{u} - \mathbf{u}' \neq 0, \|\mathbf{u} - \mathbf{u}'\|_\infty \leq 1$, then $\mathbf{u} - \mathbf{u}'$ is a solution to the ring-$\mathbf{SIS}_{m,q,1}$ problem.

Let $\mathcal{H} = \{h_{\mathbf{A}} | \mathbf{A} \in R_q^m\}$, then we give the following specific description of the dynamic updating algorithm $\mathbf{TDA}(t, \mathbf{d}^*)$ to construct and update the Merkle tree that is used to record the registered users and partial group information in this paper:

> **TSetup**: Initialize the Merkle tree as a tree with depth 1, the value of leaves are $\mathbf{0}$, and its root is $\mathbf{u}$. Let t denote the number of legal members in the group.
> **TJoin**: Search for the first leaf with value $\mathbf{0}$ in all leaves, and assume that its index is $i \leq t$. Include a tree of depth $j = \lceil \log t \rceil$ where all leaves are $\mathbf{0}$ into the original one if there is not a such leaf. And take its root $\mathbf{u}'$ and the root $\mathbf{u}$ of the original tree as two inputs of the hash function to compute a new root $\mathbf{u}_{new} = h_{\mathbf{A}}(\mathbf{u}, \mathbf{u}')$ of the new Merkle tree. And for any $i \in [2^{j+1}]$, we have $|\mathbf{bin}(i)| = j + 1$.
> **TUpdate**: Let $\mathbf{u}_{j+1} = \mathbf{d}^*$ denote the value of the leaf corresponding to the ith user, $\mathbf{bin}(i-1) = (i_1, \cdots, i_{j+1})$ is the binary description of integer $i - 1$, its witness is $w = (\mathbf{bin}(i-1), (\mathbf{w}_{j+1}, \cdots, \mathbf{w}_1))$. Update the value of notes recursively in the path $\mathbf{u}_j, \cdots, \mathbf{u}_0$ from the leaf $\mathbf{u}_{j+1}$ to root $\mathbf{u}$, then output the witness w, a new root $\mathbf{u}_{new}$, where $\mathbf{w}_{j+1}, \cdots, \mathbf{w}_1$ and $\mathbf{u}_j, \cdots, \mathbf{u}_0$ satisfy the following relationship

$$\forall l \in \{j, \cdots, 1, 0\}, \mathbf{u}_l = \begin{cases} h_{\mathbf{A}}(\mathbf{u}_{l+1}, \mathbf{w}_{l+1}), \text{ if } i_{l+1} = 0 \\ h_{\mathbf{A}}(\mathbf{w}_{l+1}, \mathbf{u}_{l+1}), \text{ if } i_{l+1} = 1 \end{cases} \quad (2)$$

> Let $\mathbf{u}_{new} = \mathbf{u}_0$ be the new root of the Merkle tree.

Given the variable t, the computational complexity of algorithm $\mathbf{TUpdate}(t, \mathbf{d}^*)$ is $O(\log t)$, and it satisfies the following property

Theorem 2 *Suppose that the problem ring-*$\mathbf{SIS}^{\infty}_{m,q,\beta}$ *is difficult, let* $R' = \{\mathbf{d}_0, \cdots, \mathbf{d}_t\}$ *be the set of the leaves related to users who have been registered, then the algorithm* $\mathbf{TDA}(t, \mathbf{d}^*)$ *is secure. And given a negligible function* $negl(\lambda)$*, for any PPT adversary* $\mathcal{A}$*, the following inequality is true*

$$\Pr[(\mathbf{d}^*, \mathbf{w}^*) \leftarrow \mathcal{A}(R', t) : \mathbf{d}^* \notin R', \mathbf{u} = \mathbf{u}_0] \leq negl(\lambda)$$

Syntax and security of fully dynamic group signature with message dependent opening

Different from the general group signature scheme, there are four participants in a fully dynamic group signature scheme with message-dependent opening(FDGS-MDO): The group manager(GM_{update}): Who is responsible to update the group information and the registration and revocation of users. The admitter(AM): who is responsible to generate a token $\mathbf{t_M}$ that specifies the signatures associated with message M would be opened. The trace manager(GM_{trace}): Given a signature and token $\mathbf{t_M}$, GM_{trace} is responsible to trace the identity of signer when there is a dispute. The users: Who are usually appeared as a signer to sign messages or a verifier to verify signatures.

The definition of FDGS-MDO

A fully dynamic group signature scheme with message-dependent opening consists of the following polynomial-time algorithms:

> $\mathbf{GKeyGen}(\lambda) \rightarrow (pp, (\mathbf{mpk}, \mathbf{msk}), (opk, osk), tsk)$: On input the security parameter λ, this algorithm outputs the public parameter pp, group public key $gpk = (pp, \mathbf{mpk}, opk)$, and the group secret key $\mathbf{msk}$ of GM_{update}, the tracing secret key osk of GM_{trace} and the secret key tsk of AM. GM_{update} initializes the registration list $\mathbf{reg}$ and the group information $\mathbf{info}$ as $\emptyset$, and we assume that they can only be edited by a party knowing $\mathbf{msk}$.
> $\mathbf{UKeyGen}(pp) \rightarrow (\mathbf{upk}, \mathbf{usk})$: Given the public parameter pp, this algorithm outputs a user's key pair $(\mathbf{upk}, \mathbf{usk})$.
> $\langle \mathbf{Join}(gpk, \mathbf{upk}), \mathbf{Issue}(gpk, \mathbf{msk}, \mathbf{reg}, \mathbf{info}) \rangle$: This algorithm is an interactive protocol between a user and the group manager GM_{update}. Assume that the new registered user is the tth member in the group, the user become a legitimate member of the group if the algorithm goes well, and the $\mathbf{Join}$ algorithm sets its signing secret key $gsk = (\mathbf{bin}(t-1), \mathbf{upk}_t, \mathbf{usk}_t)$. For the $\mathbf{Issue}$ algorithm, GM_{update} runs the algorithm $\mathbf{TDA}(t, \mathbf{upk}_t)$ to update the Merkle hash tree, the group information $\mathbf{info}_\tau$, and the registered user list $\mathbf{reg}$.
> $\mathbf{Revoke}(gpk, S, \mathbf{msk}, \mathbf{reg}, \mathbf{info}_\tau) \rightarrow \mathbf{info}_{\tau_{new}}$: Given the revocation list S, for any $i \in S$, the group manager GM_{update} runs algorithm $\mathbf{TUpdate}(\mathbf{bin}(i-1), 0^k)$ to update the Merkle hash tree, the registered user list $\mathbf{reg}$ and the group information $\mathbf{info}_{\tau_{new}}$.
> $\mathbf{Sign}(gpk, \mathbf{gsk}_i, \mathbf{info}_\tau, M) \rightarrow \Sigma$: On input group public key gpk, group information $\mathbf{info}_\tau$, this algorithm outputs a signature Σ to a message M signed by the user corresponding to ith leaf at τ or an error symbol $\perp$ if the user is illicit at τ, i.e. the user has not been registered or has been revoked at τ.
> $\mathbf{Verify}(gpk, \Sigma, \mathbf{info}_\tau, M) \rightarrow 0/1$: Verify the signature Σ and output 1 if it is valid, otherwise output 0.
> $\mathbf{TrapGen}(gpk, \mathbf{tsk}, M, \mathbf{reg}, \mathbf{info}_\tau) \rightarrow \mathbf{t_M}$: This algorithm is operated by the admitter AM, it outputs a token $\mathbf{t_M}$ for the corresponding message M.
> $\mathbf{Trace}(gpk, osk, \mathbf{t_M}, M, \Sigma, \mathbf{reg}, \mathbf{info}_\tau) \rightarrow (\mathbf{b}', \Pi_{trace})$:

This algorithm is operated by the trace manager GM_{trace}, it outputs the public key $\mathbf{b}'$ of the signer who signed the message M at τ and generate a proof for this fact if the signature Σ is valid. Otherwise output $\perp$.

Judge$(gpk, \mathbf{b}', M, \Pi_{trace}, \Sigma, \mathbf{info}_\tau) \rightarrow 0/1$: Verify the proof Π_{trace} generated by the trace manager GM_{trace}, and output 1 if it is valid, otherwise output 0.

To verify that whether the signer is legitimate or not, i.e. the signer has registered and not be revoked when he signs a message M at τ, the group manager verifies that whether the value of the leaf corresponding to this signer is non-zero. And to avoid leaking any information about the signer's identity, we use the extension-permutation technology (Libert et al. 2016) to hide it. In other words, suppose that the binary representation of the value of the leaf that corresponding to the signer is $\mathbf{bin}(\mathbf{d}_i) = (d_{i1}, d_{i2}, \cdots, d_{ik})$, $i \in [t]$, choose a vector $\mathbf{a} \xleftarrow{\$} \{0,1\}^{k-1}$ uniformly such that the Hamming weight of $\mathbf{d}_i' = (\mathbf{bin}(\mathbf{d}_i)|\mathbf{a}) \in \{0,1\}^{2k-1}$ is k. Given a random permutation $\pi_{2k-1} \in \mathcal{S}_{2k-1} = \{\pi_{2k-1} | \pi_{2k-1}$ is a random permutation of elements in $\{0,1\}^{2k-1}\}$, the Hamming weight of $\pi_{2k-1}(\mathbf{d}_i')$ is k if and only if $\mathbf{d}_i \neq 0$.

Security of FDGS-MDO scheme

A fully dynamic group signature scheme needs to satisfies the following properties: correctness, anonymity against admitter, anonymity against opener, non-frameability, traceability, and tracing soundness. Before the specific description, we would like to give a brief description of oracles and special symbols used in the proof firstly. *HUL* is the set of honest users whose secret keys are generated honesty. *BUL* is the set of users whose signing secret keys are sent to the adversary. *CUL* is the set of users whose public keys are chosen by the adversary. *SL* is the set of signatures generated by oracle **sign**. *CL* is the set of signatures generated by oracle $\mathbf{Chal}_b$, *TL* is the set of tokens generated by oracle $\mathbf{Chal}_b$. And oracles used in the proof are as follows:

AddU(i): Add an honest user i into the set *HUL* at time τ.

CreU$(i, \mathbf{upk}_i)$: Create a new user i whose public key $\mathbf{upk}_i$ is chosen by the adversary, which is invoked in the oracle **SenToM**.

SenToM(i, M_{in}): It is used to run the algorithm **Join**, on behalf of a corrupt user, together with the honest group manager GM_{update}.

SenToU(i, M_{in}): It is used to run the algorithm **Join**, on behalf of the corrupt group manager GM_{update}, together with a legitimate user i.

RReg(i): Return the registration information $\mathbf{reg}_i$ of user i.

MReg(i, ρ): Change the registration information $\mathbf{reg}_i$ of user i into ρ.

RevealU(i): Return the signing secret key gsk_i of user i to the adversary, and add i to the set *BUL*.

Sign(i, M, τ): Return a signature to a message M signed by user i at time τ, and add this signature to the set *SL*.

$\mathbf{Chal_b}(\mathbf{info}_\tau, i_0, i_1, M)$: For any $b \in \{0,1\}$, Return the signature to a message M signed by user i_b at time τ, and add this signature to the set *CL*. This requires that the users i_0, i_1 are all legitimate at time τ, and this oracle could be revoked only once.

Trace$(\mathbf{info}_\tau, \Sigma, M)$: Return the signer of a signature Σ signed at time τ and a proof of this fact, which requires that the signature $\Sigma \notin CL$.

TrapGen$(\mathbf{info}_\tau, M)$: Return a token of the message M generated at time τ, which requires that the message $M \notin TL$.

UpdateG(S, τ): It allows the adversary to update some information about the group at time τ, which requires that each element in S is legitimate user's public key at time τ.

IsActive$(\mathbf{info}_\tau, \mathbf{reg}, i)$: Return 1 if and only if the user i is a legitimate member in the group at time τ, otherwise return 0.

Correctness: This property means that if the signer signs a message M honestly, the algorithm **Verify** can always output 1. With a token $\mathbf{t}_M$ that outputted by the algorithm **TrapGen**, the trace manager GM_{trace} can trace the identity of the signer by the algorithm **Trace**, and generates a proof Π_{trace} accepted by the algorithm **Judge**.

Anonymity against admitter: For any PPT adversary $\mathcal{A}$, this property means that it is impossible to distinguish signatures generated by two legitimate users with a non-negligible probability, even though the adversary $\mathcal{A}$ could learn the secret key **msk** of GM_{update} and the secret key tsk of AM, corrupt any user, and is given the access to the oracle **Trace**. Given a negligible function $negl(\lambda)$, a DFGS-MDO scheme is anonymous against admitter for all PPT adversary $\mathcal{A}$ if $\Pr[\mathbf{Exp}^{anonA-b}_{DGS-MDO,\mathcal{A}}(\lambda) = 1] \leq negl(\lambda)$.

$(pp, (opk, osk)) \leftarrow$ **GKeyGen**(λ),
$HUL, CUL, BUL, SL, CL = \emptyset$.
$(\mathbf{info}, (\mathbf{mpk}, \mathbf{msk}), (tpk, tsk)) \leftarrow \mathcal{A}(pp)$. Return 0 if $\mathcal{A}$'s output is not well-formed, let
$gpk = (pp, \mathbf{mpk}, opk, tpk)$.
$b^* \leftarrow \mathcal{A}^{\mathbf{AddU},\mathbf{CreU},\mathbf{RevealU},\mathbf{SenToU},\mathbf{Trace},\mathbf{MReg},\mathbf{Chal_b}}(gpk)$,
return b^*.

Anonymity against opener: For any PPT adversary $\mathcal{A}$, this property means that it is impossible to distinguish signatures generated by two legitimate users with a non-

negligible probability, even though the adversary $\mathcal{A}$ could learn the secret key **msk** of GM_{update} and the secret key *tsk* of AM, corrupt any user, and is given the access to the oracle **TrapGen**. Given a negligible function $negl(\lambda)$, a DFGS-MDO scheme is anonymous against opener for all PPT adversary $\mathcal{A}$ if $\Pr[\mathbf{Exp}^{anonO-b}_{DGS-MDO,\mathcal{A}}(\lambda)=1] \le negl(\lambda)$.

$(pp,(tpk,tsk)) \leftarrow \mathbf{GKeyGen}(\lambda)$,
$HUL, CUL, BUL, SL, TL = \emptyset$.
$(\mathbf{info},(\mathbf{mpk},\mathbf{msk}),(opk,osk)) \leftarrow \mathcal{A}(pp)$. Return 0 if $\mathcal{A}$'s output is not well-formed, let
$gpk = (pp,\mathbf{mpk},opk,tpk)$. $b^* \leftarrow \mathcal{A}^{\mathbf{AddU},\mathbf{CreU},\mathbf{RevealU},\mathbf{SenToU},\mathbf{TrapGen},\mathbf{MReg},\mathbf{Chal_b}}(gpk)$,
return b^*.

Non-frameability: For any PPT adversary $\mathcal{A}$, the probability to generate a valid signature that traced to a legitimate user is negligible, even though the adversary $\mathcal{A}$ could learn the secret keys of GM_{update} and GM_{trace}, and corrupt some of the users. Given a negligible function $negl(\lambda)$, a DFGS-MDO scheme satisfies non-frame-ability for all PPT adversary $\mathcal{A}$ if $\Pr[\mathbf{Exp}^{unforge}_{FDGS-MDO,\mathcal{A}}(\lambda)=1] \le negl(\lambda)$.

$pp \leftarrow \mathbf{GKeyGen}(\lambda), HUL, CUL, BUL, SL = \emptyset$.
$(\mathbf{info},(\mathbf{mpk},\mathbf{msk}),(opk,osk),(tpk,tsk)) \leftarrow \mathcal{A}(pp)$.
return 0 if $\mathcal{A}$'s output is not well-formed, let
$gpk = (pp,\mathbf{mpk},opk,tpk)$. $(M,\Sigma,i,\Pi_{trace},\mathbf{info}_\tau) \leftarrow \mathcal{A}^{\mathbf{CreU},\mathbf{RevealU},\mathbf{SenToU},\mathbf{Sign}}(gpk)$. return 1 if
$\mathbf{Verify}(gpk,\mathbf{info}_\tau,M,\Sigma) =$
$1 \wedge \mathbf{Judge}(gpk,\mathbf{upk}_i,\mathbf{info}_\tau,\Pi_{trace},M,\Sigma)=1 \wedge i \in HUL \setminus BUL \wedge (M,\Sigma,\tau) \notin SL$.

Traceability: For any PPT adversary $\mathcal{A}$, the probability to generate a valid signature that traced to a illicit user is negligible, even though the adversary $\mathcal{A}$ could learn the secret key of GM_{trace} and corrupt some of the users. Given a negligible function $negl(\lambda)$, a DFGS-MDO scheme is traceable for all PPT adversary $\mathcal{A}$ if $\Pr[\mathbf{Exp}^{trace}_{FDGS-MDO,\mathcal{A}}(\lambda)=1] \le negl(\lambda)$.

$(pp,(\mathbf{mpk},\mathbf{msk})) \leftarrow \mathbf{GKeyGen}(\lambda)$,
$HUL, CUL, BUL, SL = \emptyset$.
$(\mathbf{info},(opk,osk),(tpk,tsk)) \leftarrow \mathcal{A}(pp)$. return 0 if $\mathcal{A}$'s output is not well-formed, let
$gpk = (pp,\mathbf{mpk},opk,tpk)$. $(M,\Sigma,\tau) \leftarrow \mathcal{A}^{\mathbf{AddU},\mathbf{CreU},\mathbf{SenToM},\mathbf{RevealU},\mathbf{MReg},\mathbf{Sign},\mathbf{UpdateG}}(gpk)$.
return 0 if $\mathbf{Verify}(gpk,\mathbf{info}_\tau,M,\Sigma)=0$.
$(i,\Pi_{trace}) \leftarrow \mathbf{Trace}(gpk,osk,\mathbf{t_M},\mathbf{info}_\tau,\mathbf{reg},M,\Sigma)$.
return 1 if $\mathbf{IsActive}(\mathbf{info}_\tau,\mathbf{reg},i) = \perp$
$\vee \mathbf{Judge}(gpk,\mathbf{upk}_i,\mathbf{info}_\tau,\Pi_{trace},M,\Sigma)=0 \vee i=0$.

Tracing soundness: For any PPT adversary $\mathcal{A}$, the probability to generate a valid signature that traced to two different users is negligible, even though the adversary $\mathcal{A}$ could learn the secret keys of GM_{update} and GM_{trace}, and corrupt some of the users. Given a negligible function $negl(\lambda)$, a DFGS-MDO scheme satisfies tracing soundness for all PPT adversary $\mathcal{A}$ if $\Pr[\mathbf{Exp}^{trace-sound}_{FDGS-MDO,\mathcal{A}}(\lambda)=1] \le negl(\lambda)$.

$pp \leftarrow \mathbf{GKeyGen}(\lambda)$, $CUL = \emptyset$.
$(\mathbf{info},(\mathbf{mpk},\mathbf{msk}),(opk,osk),(tpk,tsk)) \leftarrow \mathcal{A}(pp)$.
return 0 if $\mathcal{A}$'s output is not well-formed, let
$gpk = (pp,\mathbf{mpk},opk,tpk)$.
$(M,\Sigma,i_0,\Pi_{trace,i_0},i_1,\Pi_{trace,i_1},\mathbf{info}_\tau) \leftarrow \mathcal{A}^{\mathbf{CreU},\mathbf{MReg}}(gpk)$. return 1 if for $b \in \{0,1\}$,
$\mathbf{Verify}(gpk,\mathbf{info}_\tau,M,\Sigma)=1 \wedge i_0 \neq i_1 \neq \perp \wedge$
$\mathbf{Judge}(gpk,\mathbf{upk}_{i_b},\mathbf{info}_\tau,\Pi_{trace},M,\Sigma)=1$.

The lattice-based dynamic group signature scheme with message-dependent opening

The construction of the scheme

By using the dynamic algorithm to construct the Merkle hash tree and the formal definition of the fully dynamic group signature scheme with message-dependent opening, the specific construction of the scheme in this paper could be defined as follows:

$\mathbf{GKeyGen}(\lambda) \rightarrow (pp,(\mathbf{mpk},\mathbf{msk}),(opk,osk),tsk)$: Given the security parameter λ, let $t > 0$ denote the number of registered users, $l = \lceil \log t \rceil$, $n = O(\lambda)$, $p = O(\lambda)$, prime modules $q = \tilde{O}(n^{1.5})$, $k = n\lceil \log q \rceil$, $m = 2k$, real integer $\beta > 0$, $\mathcal{X}$ is the noise distribution bounded by β in R, $H : \{0,1\}^* \rightarrow \{0,1\}^{k'}$ is a hash function for FS transformation, $H' : \{0,1\}^* \rightarrow \mathcal{X}^k$ is a collision resistant hash function, and $Com : \{0,1\}^* \times \{0,1\}^m \rightarrow \mathbb{Z}_q^n$ is a string commitment scheme with properties of statistical hiding and computational binding (Kawachi et al. 2008). Let $\mathbf{A} \xleftarrow{\$} R_q^m$, $\mathbf{B} \xleftarrow{\$} R_q^k$. GM_{update} chooses $\mathbf{msk} \xleftarrow{\$} \{0,1\}^m$, computes $\mathbf{mpk} = \mathbf{A} \cdot \mathbf{msk}$, and initializes the registration list **reg** and the group information **info** as $\emptyset$. GM_{trace} chooses $\mathbf{S}_1, \mathbf{S}_2 \xleftarrow{\$} \mathcal{X}^k$, $E_1, E_2 \xleftarrow{\$} \mathcal{X}$, and computes $P_1 = \mathbf{S}_1^\top \mathbf{B} + E_1 \in R_q$, $P_2 = \mathbf{S}_2^\top \mathbf{B} + E_2 \in R_q$. AM chooses $\mathbf{S}_3, \mathbf{S}_4 \xleftarrow{\$} \mathcal{X}^k$, $E_3, E_4 \xleftarrow{\$} \mathcal{X}$. Set the GM_{trace}'s key pair $(opk,osk) = (P_1,(\mathbf{S}_1,E_1))$, the GM_{update}'s key pair $(\mathbf{mpk},\mathbf{msk})$, and the AM's secret key $tsk = (\mathbf{S}_3,E_3)$. Finally, the algorithm outputs the public parameter $pp =$

$(\lambda, n, p, q, k, m, \beta, \mathcal{X}, H, Com, \mathbf{A}, \mathbf{B})$, the group public key $gpk = (pp, \mathbf{mpk}, opk, tpk)$.

UKeyGen$(pp) \rightarrow (\mathbf{upk}, \mathbf{usk})$: The user chooses $\mathbf{usk} \xleftarrow{\$} \{0,1\}^m$ uniformly as its secret key, and computes the related public key $\mathbf{upk} = \mathbf{bin}(\mathbf{A} \cdot \mathbf{usk} \bmod q)$, and $\mathbf{upk} \in \{0,1\}^k$.

$\langle \mathbf{Join}(gpk, \mathbf{upk}), \mathbf{Issue}(gpk, \mathbf{msk}, \mathbf{reg}, \mathbf{info}) \rangle$: Assume that the new registered user is the t-th member in the group, and the user sends its public key **upk** to the group manager GM_{update}, and if this algorithm goes well, the latter searches and denotes the first non-zero leaf as t' if he approves the user's application. Let $\mathbf{upk}_{t'} = \mathbf{upk}$, $\mathbf{reg}_{t'} = \mathbf{reg}_{t'}[\mathbf{upk}_{t'}][\tau]$, τ is the time the user registered, GM_{update} includes $\mathbf{reg}_{t'}$ into the registration list $\mathbf{reg} := (\mathbf{reg}_1[\mathbf{upk}_1][\tau], \cdots, \mathbf{reg}_{t'}[\mathbf{upk}_{t'}][\tau], \cdots, \mathbf{reg}_t[\mathbf{upk}_t][\tau])$. Then GM_{update} ru-ns the algorithm $\mathbf{TDA}(\mathbf{bin}(t'), \mathbf{upk}_{t'})$ to update the Merkle tree, outputs the group information $\mathbf{info}_\tau = (\mathbf{u}, \{\mathbf{w}_j\}_{i_j})$ where $\mathbf{u}$ is the root and $\{\mathbf{w}_j\}_{i_j}$ are witnesses of all legal users, and updates the counter of registered users $t = t + 1$. Let $\mathbf{usk}_{t'} = \mathbf{usk}$, the user sets $gsk_{t'} = (\mathbf{bin}(t' - 1), \mathbf{upk}_{t'}, \mathbf{usk}_{t'})$ as its signing secret key.

Revoke$(gpk, S, \mathbf{msk}, \mathbf{reg}, \mathbf{info}_\tau) \rightarrow \mathbf{info}_{\tau_{new}}$: Given the revocation list S that is the set of public keys of group members who would be revoked, and if $S = \{\mathbf{upk}_{i_1}, \cdots, \mathbf{upk}_{i_r}\}$ is not empty, where $r \geq 1$, $i_j \in [t]$, $j \in [r]$, for every $j \in [r]$, $\mathbf{upk}_{i_j} \in S$, GM_{update} runs the algorithm **TUpdate** in $TDA(\mathbf{bin}(i_j - 1), 0^k)$ to update the Merkle hash tree, then updates the registration list **reg**: changes $\mathbf{reg}_{i_j}[\mathbf{upk}_{i_j}][\tau]$ to $\mathbf{reg}_{i_j}[0^k][\tau_{new}]$ if $\mathbf{upk}_{i_j} \in S$, otherwise changes $\mathbf{reg}_{i_j}[\mathbf{upk}_{i_j}][\tau]$ to $\mathbf{reg}_{i_j}[\mathbf{upk}_{i_j}][\tau_{new}]$, finally outputs the new group information $\mathbf{info}_{\tau_{new}} = (\mathbf{u}_{new}, \{\mathbf{w}_j\}_{i_j})$ that consists of a new root $\mathbf{u}_{new}$ and witnesses $\{\mathbf{w}_j\}_{i_j}$ of $\mathbf{upk}_{i_j}$, updates the counter of legitimate users $t = t - r$. So, the leaves with value 0^k in the Merkle tree corresponding to the potential users who have not been registered or those have been revoked.

Sign$(gpk, \mathbf{gsk}_i, \mathbf{info}_\tau, M) \rightarrow \Sigma$: To sign a message M at τ by using the group information $\mathbf{info}_\tau$, the user related to the ith leaf verifies that whether there is a witness of $\mathbf{bin}(i - 1)$ in $\mathbf{info}_\tau$ firstly, if not, return $\perp$. Otherwise, the user sends M to AM, receives $P_3 = \tilde{\mathbf{S}}_3^\top \mathbf{B} + E_3$ and $P_4 = \tilde{\mathbf{S}}_4^\top \mathbf{B} + E_4$ from it, where $\tilde{\mathbf{S}}_3 = H'(\mathbf{S}_3 \| M)$, $\tilde{\mathbf{S}}_4 = H'(\mathbf{S}_4 \| M)$, and obtains $(\mathbf{bin}(i - 1), (\mathbf{w}_l, \cdots, \mathbf{w}_1))$ from $\mathbf{info}_\tau$ to do the follows: Choose random strings $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4 \xleftarrow{\$} \{0,1\}^k$, the user encrypts vector $\mathbf{upk}_i$ by making use of the double-encryption paradigm (Naor and Yung 1990) and the RLWE-based encryptionp scheme (Regev 2009; Lyubashevsky et al. 2013) to obtain the ciphertexts,

$$\begin{aligned}\mathbf{c}_1 &= (c_{1,1}, \mathbf{c}_{1,2}) \\ &= \left(\mathbf{B} \cdot \mathbf{r}_1 \mod q, P_1 \cdot \mathbf{r}_1 + \left\lceil \frac{q}{2} \right\rfloor \cdot \mathbf{upk}_i \mod q\right) \in R_q \times R_q^k,\end{aligned}$$

$$\begin{aligned}\mathbf{c}_2 &= (c_{2,1}, \mathbf{c}_{2,2}) \\ &= \left(\mathbf{B} \cdot \mathbf{r}_2 \mod q, P_2 \cdot \mathbf{r}_2 + \left\lceil \frac{q}{2} \right\rfloor \cdot \mathbf{upk}_i \mod q\right) \in R_q \times R_q^k.\end{aligned}$$

Then encrypt ciphertext $\mathbf{c}_{1,2}$ by using a method similar to the one above to obtain the ciphertexts,

$$\begin{aligned}\mathbf{c}_3 &= (c_{3,1}, \mathbf{c}_{3,2}) \\ &= \left(\mathbf{B} \cdot \mathbf{r}_3 \mod q, P_3 \cdot \mathbf{r}_3 + \left\lceil \frac{q}{2} \right\rfloor \cdot \mathbf{c}_{1,2} \mod q\right) \in R_q \times R_q^k,\end{aligned}$$

$$\begin{aligned}\mathbf{c}_4 &= (c_{4,1}, \mathbf{c}_{4,2}) \\ &= \left(\mathbf{B} \cdot \mathbf{r}_4 \mod q, P_4 \cdot \mathbf{r}_4 + \left\lceil \frac{q}{2} \right\rfloor \cdot \mathbf{c}_{1,2} \mod q\right) \in R_q \times R_q^k.\end{aligned}$$

Finally, the signer generates a non-interactive zero-knowledge argument of knowledge(NIZKAoK) Π_{sign} for:

(1) It has legitimate witness $\zeta = (\mathbf{usk}_i, \mathbf{upk}_i, \mathbf{bin}(i), \mathbf{w}_l, \cdots, \mathbf{w}_1, \mathbf{r}_1, \cdots, \mathbf{r}_4)$ such that the signer is a legitimate member in the group, i.e. $\mathbf{upk}_i \neq 0^k$, and the values of nodes in the path that from the leaf corresponding to the user to the root are all correct.
(2) $(\mathbf{usk}_i, \mathbf{upk}_i)$ is a valid public-private key-pair.
(3) $(\mathbf{c}_1, \mathbf{c}_2)$ are two legitimate ciphertext of $\mathbf{upk}_i$.
(4) $(\mathbf{c}_3, \mathbf{c}_4)$ are two legitimate ciphertext of $\mathbf{c}_{1,2}$.

Output the signature $\Sigma = (c_{1,1}, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \Pi_{sign})$. The NIZKAoK mentioned above is obtained from the interactive protocol in the latter section by FS transformation, i.e. runs the underlying protocol $k' = \lceil \frac{\lambda}{\log_2(\max(n,p)+1)} \rceil$ times sequentially to obtain a negligible soundness error $2^{-\lambda}$, and the transcript is $\Pi_{sign} = \left(\{(\mathbf{u}_j, \hat{\mathbf{u}}_j)\}_{j=1}^{k'}, \mathbf{ch}, \{rsp_j\}_{j=1}^{k'}\right)$, where

$$\begin{aligned}\mathbf{ch} &= (ch_1, \cdots, ch_{k'}) \in ([n] \times \mathbb{Z}_p)^{k'} \\ &= H\left(M, \{(\mathbf{u}_j, \hat{\mathbf{u}}_j)\}_{j=1}^{k'}, \mathbf{A}, \mathbf{u}_\tau, \mathbf{B}, \{P_i\}_{i=1}^4, c_{1,1}, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4\right)\end{aligned}$$

Verify$(gpk, \Sigma, \mathbf{info}_\tau, M) \rightarrow 0/1$: The verifier obtains the root $\mathbf{u}_\tau$ of the Merkle hash tree at τ from the group information $\mathbf{info}_\tau$, and verifies that whether the predicted challenge **ch** is true, outputs 0 if not, otherwise verifies the respond rsp_j that corresponding to $(\mathbf{u}_j, \hat{\mathbf{u}}_j)$ and ch_j for each $j \in [k']$, and outputs 1 if everything is correct, otherwise outputs 0.

TrapGen$(gpk, \mathbf{tsk}, M, \mathbf{reg}, \mathbf{info}_\tau) \rightarrow \mathbf{t_M}$: If a token $\mathbf{t_M}$ for message M was already queried, answer consistently. Otherwise, compute $\tilde{\mathbf{S}}_3 = H'(\mathbf{S}_3 \| M)$, let $\mathbf{t_M} = (\tilde{\mathbf{S}}_3, E_3)$, and outputs $\mathbf{t_M}$.

Trace$(gpk, osk, \mathbf{t_M}, M, \Sigma, \mathbf{reg}, \mathbf{info}_\tau) \rightarrow (\mathbf{b}', \Pi_{trace})$: Firstly, trace manager GM_{trace} uses token $\mathbf{t_M}$ to decrypt ciphertext $\mathbf{c}_3$ to get $\mathbf{c}'_{1,2}$, i.e. computes $\mathbf{c}'_{1,2} = \left\lfloor \frac{(\mathbf{c}_{3,2} - \tilde{S}_3^\top \cdot c_{3,1})}{q/2} \right\rceil \in \{0,1\}^k$, and the ciphertexts $\mathbf{c}_2$ and $\mathbf{c}_4$ are only used in our proof. Let $c'_{1,1} = c_{1,1}$, then GM_{trace} uses its tracing secret key osk to decrypt the ciphertext $\mathbf{c}'_1 = (c'_{1,1}, \mathbf{c}'_{1,2})$ and computes $\mathbf{b}' = \left\lfloor \frac{(\mathbf{c}'_{1,2} - S_1^\top \cdot c'_{1,1})}{q/2} \right\rceil \in \{0,1\}^k$. If there is not a witness of $\mathbf{b}'$ in $\mathbf{info}_\tau$ or $\mathbf{b}' = 0^k$, output $\perp$. Then GM_{trace} generates a non-interactive zero-knowledge argument of knowledge(NIZKAoK) Π_{trace} for the fact that the user corresponding to $\mathbf{b}'$ really generated a signature Σ to message M at τ. In other words, the trace manager GM_{trace} should proof that he has $\mathbf{t_M} = (\tilde{\mathbf{S}}_3, E_3)$, $\mathbf{S}_1, \tilde{\mathbf{S}}_3 \in R_q^k$, $E_1, E_3 \in R_q$, $\mathbf{y}_1, \mathbf{y}_3 \in R_q^k$, such that

$$\begin{aligned}
\|\mathbf{S}_1\|_\infty, \|\tilde{\mathbf{S}}_3\|_\infty \le \beta, |E_1|, |E_3| \le \beta, \|\mathbf{y}_1\|_\infty, \|\mathbf{y}_3\|_\infty &\le \left\lceil \frac{q}{5} \right\rceil \\
\mathbf{S}_1^\top \cdot \mathbf{B} + E_1 &= P_1 \mod q \\
\tilde{\mathbf{S}}_3^\top \cdot \mathbf{B} + E_3 &= P_3 \mod q \\
\mathbf{c}_{3,2} - \tilde{\mathbf{S}}_3^\top \cdot c_{3,1} &= \mathbf{y}_3 + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathbf{c}'_{1,2} \mod q \\
\mathbf{c}'_{1,2} - \mathbf{S}_1^\top \cdot c_{1,1} &= \mathbf{y}_1 + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathbf{b}' \mod q
\end{aligned}$$

Similarly, the NIZKAoK mentioned above is obtained from the interactive protocol in the latter section by FS transformation, i.e. GM_{trace} runs the underlying protocol $k' = \left\lceil \frac{\lambda}{\log_2(\max(n,p)+1)} \right\rceil$ times sequentially to obtain a negligible soundness error $2^{-\lambda}$, and the transcript is $\Pi_{trace} = \left(\{(\mathbf{u}_j, \hat{\mathbf{u}}_j)\}_{j=1}^{k'}, \mathbf{ch}, \{rsp_j\}_{j=1}^{k'} \right)$, where $\mathbf{ch} \in ([n] \times \mathbb{Z}_p)^{k'}$,

$$\mathbf{ch} = (ch_1, \cdots, ch_{k'}) = H\left(M, \{(\mathbf{u}_j, \hat{\mathbf{u}}_j)\}_{j=1}^{k'}, gpk, \Sigma, \mathbf{info}_\tau, \mathbf{t_M}, \mathbf{b}'\right)$$

Finally, this algorithm outputs $(\mathbf{b}', \Pi_{trace})$.

Judge$(gpk, \mathbf{b}', M, \Pi_{trace}, \Sigma, \mathbf{info}_\tau) \rightarrow 0/1$: Verify the proof Π_{trace} and output 1 if it is true, otherwise output 0.

Finally, a timestamp τ is given to each member in the group, the group manager GM_{update} updates the group information $\mathbf{info}_\tau$ once a new user registered or a legitimate member has been revoked, which indicates that the user can not sign a message M before a registration or after a revocation. Given a group information $\mathbf{info}_\tau$, we can confirm the timestamp τ uniquely, and vice versa. For any two timestamps $\tau_1 < \tau_2$, the group information $\mathbf{info}_{\tau_1}$ is published earlier than $\mathbf{info}_{\tau_2}$.

Analysis of the lattice-based FDGS-MDO scheme

In our scheme, it is not necessary to prepare a large storage space for the Merkle tree standby before a signature is generated, namely we only need to extend or update the Merkle hash tree when a user needs a registration or be revoked. Compared with the scheme in Ling et al. (2017), our work could economize considerable storage space, and there is also no limits on the upper bound of the size of the group as long as the storage space is allowed. In addition, the fact that the scheme is implemented based on ring could help to reduce the computational complexity and space complexity of it (Table 1).

Complexity: Given a security parameter λ, the size of legitimate users t, $l = \lceil \log t \rceil$, $n = O(\lambda)$, $q = \tilde{O}(n^{1.5}) = \tilde{O}(c\lambda^{1.5})$ with a constant c, $k = n\lceil \log q \rceil = O(\lambda \log \lambda)$. Then the size of group public key $gpk = (pp, \mathbf{mpk}, opk, tpk)$ is $|gpk| = O(nk) + k + O(k^2) = O((\lambda \log \lambda)^2)$, the size of signing secret key $gsk_i = (\mathbf{bin}(i), \mathbf{upk}_i, \mathbf{usk}_i)$ is $|gsk_i| = l + 3k = l + O(\lambda \log \lambda) = l + \tilde{O}(\lambda)$, and the size of signature $\Sigma = (c_{1,1}, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \Pi_{sign})$ is

Table 1 Comparison of lattice-based group signature schemes in (Libert et al. 2016) and (Ling et al. 2017), in terms of efficiency and functionality

Schemes	Security level	Signature size	Group PK size	Signer's SK size	Trapdoor?	Model
(Libert et al. 2016)	$\left(\frac{2}{3}\right)^{\omega(\lambda)}$	$\tilde{O}(\lambda \cdot l)$	$\tilde{O}(\lambda^2 \cdot l)$	$\tilde{O}(\lambda)$	yes	MDO
(Ling et al. 2017)	$\left(\frac{2}{3}\right)^{\omega(\lambda)}$	$\tilde{O}(\lambda \cdot l)$	$\tilde{O}(\lambda^2 + \lambda \cdot l)$	$\tilde{O}(\lambda) + l$	free	fully dynamic
Ours	$2^{-\lambda}$	$O(l\lambda^2)$	$O((\lambda \log \lambda)^2)$	$\tilde{O}(\lambda) + l$	free	fully dynamic/MDO

The scheme in (Libert et al. 2016) is static and that in (Ling et al. 2017) is fully dynamic, the similarity is that both of them use the Stern-like protocol with a soundness error $\frac{2}{3}$ as the underlying protocol. The scheme in this paper is fully dynamic and use a more efficient zero knowledge protocol with a soundness error $\frac{1}{\max(n,p)+1}$ as the underlying protocol. Obviously, compared with the previous two schemes, our scheme has lower computational complexity when realize the same security level

$$
\begin{aligned}
|\Sigma| =& |\Pi_{sign}| + |c_{1,1}| + |\mathbf{c}_2| + |\mathbf{c}_3| + |\mathbf{c}_4| \\
=& k' \cdot (|(\mathbf{u}_j, \hat{\mathbf{u}}_j)| + |ch_j| + \cdot |rsp_j|) + 1 + 3(k+1)\log q \\
=& k' \cdot (2k + \log p + \log n + 2(\log q + k \log k) + 2\lambda + D) \\
& + 1 + 3(k+1)\log q \\
=& O(l\lambda^2)
\end{aligned}
$$

The soundness error of our underlying protocol is $\frac{1}{\max(n,p)+1}$, so we need to perform the protocol $\frac{\lambda}{\log(\max(n,p)+1)}$ times sequentially to reach a negligible soundness error $2^{-\lambda}$, and the generated group signature size is $O(l\lambda^2)$. To realize the same soundness error, the underlying protocol in Ling et al. (2017) need to be excluded $\Theta(\lambda)$ times sequentially, and the corresponding group signature size would be $\tilde{O}(l\lambda^2)$. Let the upper bounds of the size of the group in (Ling et al. 2017) and that in our work are the same and denoted as N, let $l = \log N$, then the expected computational complexity of realizing the dynamic registration and revocation of the counterpart of the scheme in Ling et al. (2017) over ring is $O(l)$, and that of our work is roughly $\frac{1}{2}O(l)$, So the expected computational complexity down almost by half. Correspondingly, the space complexity has been reduced by the same magnitude.

The security of the fully dynamic group signature scheme presented in this paper satisfies some security requirements given in Bootle et al. (2016): correctness, anonymity, non-frameability, traceability, and tracing soundness.

Correctness: Now, we give a specific description of the correctness of our scheme according to the perfect completeness of the underlying protocol and the correctness of the encryption scheme. If the signature $\Sigma = (c_{1,1}, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \Pi_{sign})$ is generated by a legitimate user, then the perfect completeness of the underlying protocol could help the signature Σ to pass the verification of the algorithm **Verify**, and the algorithm **Trace** will take the token $\mathbf{t_M}$ outputted by the algorithm **TrapGen** as one of the inputs to decrypt the ciphertext $\mathbf{c}_3$ and outputs $\mathbf{c}_{1,2}$, then let $\mathbf{c}_1 = (c_{1,1}, \mathbf{c}_{1,2})$, and uses its secret key *osk* to decrypt $\mathbf{c}_1$ and outputs the user public key $\mathbf{b}' = \mathbf{upk}_i$ with a probability approximate to 1 together with a proof Π_{trace} accepted by **Judge**. We need to compute $\mathbf{e}_1 = \mathbf{c}_{3,2} - \tilde{\mathbf{S}}_3^{\top} c_{3,1} = E_3 \cdot \mathbf{r}_3 + \lfloor \frac{q}{2} \rfloor \cdot \mathbf{c}_{1,2} \mod q$ and $\mathbf{e}_2 = \mathbf{c}_{1,2} - \mathbf{S}_1^{\top} c_{1,1} = E_1 \cdot \mathbf{r}_1 + \lfloor \frac{q}{2} \rfloor \cdot \mathbf{upk}_i \mod q$ when to decrypt a ciphertext, and for $s = 1, 2$, let $\mathbf{b}'_s = (b'_{s,1}, \cdots, b'_{s,l})$, $\mathbf{e}_s = (e_{s,1}, \cdots, e_{s,l})$, for any $j \in [l]$,

$$
b'_{s,j} = \begin{cases} 0, \ if \ 0 < |e_{s,j}| < \frac{q}{2} \\ 1, \ if \ \frac{q}{2} < |e_{s,j}| \end{cases} \tag{3}
$$

Note that $\|E_{s'} \cdot \mathbf{r}_{s'}\|_\infty < \frac{q}{5}$ for $s' = 1, 3$, so $\mathbf{b}'_1 = \mathbf{c}_{1,2}$, $\mathbf{b}'_2 = \mathbf{upk}_i$ with overwhelming probability. Furthermore, because the user corresponding to $\mathbf{upk}_i$ is legitimate, then the witness $w = (\mathbf{bin}(i-1), \mathbf{w}_l, \cdots, \mathbf{w}_1)$ is included in the group information $\mathbf{info}_\tau$, and the value of the related leaf is not 0^k. So, the algorithm **Trace** could always obtain a tuple $(\mathbf{S}_1, E_1, \mathbf{y}, \mathbf{t_M})$ that satisfies requirement. And finally, for the fact that the proof Π_{trace} is perfect completeness, the algorithm **Judge** outputs 1 with probability 1.

Theorem 3 *The FDGS-MDO scheme satisfies anonymous against admitter, anonymous against opener, unforgeable, traceable and tracing soundness security requirements under the ring-*$\mathbf{LWE}_{n,m,q,\mathcal{X}}$ *and ring-*$\mathbf{SIS}^{\infty}_{n,m,q,1}$ *assumptions in RO model.*

The proof of Theorem in "The improved zero-knowledge protocol of knowledge" section consists of the following five lemmas.

Lemma 1 *Suppose that the ring-*$\mathbf{LWE}_{n,m,q,\mathcal{X}}$ *problem is difficult, then the scheme in this paper is anonymous against admitter in RO model.*

Proof Assume that the size of legitimate users is t, the adversary $\mathcal{A}$ and challenger $\mathcal{C}$ are all PPT algorithms. For two different users $i_0 \neq i_1 \in [t]$ given by $\mathcal{A}$,

we say that the scheme satisfies anonymity if there is a negligible function $negl(\lambda)$, such that $\Pr\left[\mathbf{Exp}^{anonA-b}_{DGS-MDO,\mathcal{A}}(\lambda) = 1\right] \leq negl(\lambda)$. Given a negligible function $negl(\lambda)$, we will finish this proof by hybrid games. Let the output of each game is OP_l, $l \in [9]$.

Game0: Given two different legitimate users $i_0 \neq i_1 \in [t]$ by $\mathcal{A}$, let $b = 0$, the challenger $\mathcal{C}$ runs the experiment $\mathbf{Exp}^{anonA-b}_{DGS-MDO,\mathcal{A}}(\lambda)$ honestly by using i_0.

Game1: This game is completely consistent with **Game0** except that include $(\mathbf{S}_2, E_2)$ to *osk*, i.e. let $osk = ((\mathbf{S}_1, E_1), (\mathbf{S}_2, E_2))$. And this change, to the view of the adversary $\mathcal{A}$, makes no difference, $\Pr[OP_1 = 1] = \Pr[OP_0 = 1]$.

Game2: This game is completely consistent with **Game1** except that use a simulator Sim_{trace} to simulate the real interactions of the protocol that generates Π_{trace}, i.e. replace the real transcript Π_{trace} with a simulated transcript of Sim_{trace}. And the two transcripts are statistical indistinguishable because of the statistical zero-knowledge of Π_{trace}, $\Pr[OP_2 = 1] - \Pr[OP_1 = 1] \leq negl(\lambda)$.

Game3: This game is completely consistent with **Game2** except that replace $(\mathbf{S}_1, E_1)$ with $(\mathbf{S}_2, E_2)$ when Sim_{trace} simulates the oracle **Trace**. For a legitimate signature $(c_{1,1}, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \Pi_{sign})$, where $\mathbf{c}_1, \mathbf{c}_2$ are encryptions to different strings respectively, let F_1 be a event of the above signature inquiry initiated by $\mathcal{A}$ to the oracle **Trace**, and the view of $\mathcal{A}$ may changing if F_1 appears, however, it violates the soundness of the protocol that generates

Π_{sign}. And the change in this game, to the view of $\mathcal{A}$, is indistinguishable except the incident F_1, i.e. $\Pr[OP_3 = 1] - \Pr[OP_2 = 1] \leq \Pr[F_1] \leq negl(\lambda)$.

Game4: This game is completely consistent with **Game3** except that use a simulator Sim_{sign} to simulate the real interactions of the protocol that generates Π_{sign}, i.e. replace the real transcript Π_{sign} with a simulated transcript of Sim_{sign}. And the two transcripts are statistical indistinguishable because of the statistical zero-knowledge of Π_{sign}, $\Pr[OP_4 = 1] - \Pr[OP_3 = 1] \leq negl(\lambda)$.

Game5: This game is completely consistent with **Game4** except that change the ciphertext $\mathbf{c}_1$ into the encryption to $\mathbf{upk}_{i_1}$ when initiate an inquiry to the oracle $\mathbf{Chal}_b$. And the difference of the view of $\mathcal{A}$ caused by this change is negligible for the semantic security of the encryption scheme. The challenger responds with $(\mathbf{S}_2, E_2)$ during the inquiry to the oracle **Trace**, which makes no difference by substitute the ciphertext $\mathbf{c}_1$, so, $\Pr[OP_5 = 1] - \Pr[OP_4 = 1] = negl(\lambda)$.

Game6: This game is completely consistent with **Game5** except that replace $(\mathbf{S}_2, E_2)$ with $(\mathbf{S}_1, E_1)$ when Sim_{trace} simulates the oracle **Trace**. For a legitimate signature $(c_{1,1}, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4, \Pi_{sign})$, where $\mathbf{c}_1, \mathbf{c}_2$ are encryptions to different strings respectively, let F_2 be a event of the above signature inquiry initiated by $\mathcal{A}$ to the oracle **Trace**, which violates the simulation soundness of the protocol that generates Π_{sign}. And the change in this game, to the view of $\mathcal{A}$, is indistinguishable except the incident F_2, $\Pr[OP_6 = 1] - \Pr[OP_5 = 1] \leq \Pr[F_2] \leq negl(\lambda)$.

Game7: This game is completely consistent with **Game6** except that change the ciphertext $\mathbf{c}_2$ into the encryption to $\mathbf{upk}_{i_1}$. And the difference of the view of $\mathcal{A}$ caused by this change is negligible for the semantic security of the encryption scheme. The challenger responds with $(\mathbf{S}_1, E_1)$ during the inquiry to the oracle **Trace**, so change $\mathbf{c}_2$ makes no difference to the view of the adversary, $\Pr[OP_7 = 1] - \Pr[OP_6 = 1] = negl(\lambda)$.

Game8: This game is completely consistent with **Game7** except that replace the simulator Sim_{sign} with a real protocol that generates Π_{sign}, i.e. replace the simulated transcript of Sim_{sign} by a real transcript Π_{sign}. And the two transcripts are statistical indistinguishable because of the statistical zero knowledge of the protocol Π_{sign}, $\Pr[OP_8 = 1] - \Pr[OP_7 = 1] \leq negl(\lambda)$.

Game9: This game is completely consistent with **Game8** except that replace the simulator Sim_{trace} with a real protocol that generates Π_{trace}, i.e. replace the simulated transcript of Sim_{trace} by a real transcript Π_{trace}. And the two transcripts are statistical indistinguishable because of the statistical zero knowledge of the protocol Π_{trace}, $\Pr[OP_9 = 1] - \Pr[OP_8 = 1] \leq negl(\lambda)$.

Finally, we could learn from the games above that the probability:

$$\begin{aligned}&\Pr[OP_9 = 1] - \Pr[OP_0 = 1]\\ =&\Pr\left[\mathbf{Exp}^{anonA-1}_{FDGS-MDO,\mathcal{A}}(\lambda)\right] - \Pr\left[\mathbf{Exp}^{anonA-0}_{FDGS-MDO,\mathcal{A}}(\lambda)\right]\\ \leq& c \cdot negl(\lambda)\end{aligned}$$

where c is a constant. So, the scheme satisfies the property of anonymity against admitter. □

Lemma 2 *Suppose that the ring-*$\mathbf{LWE}_{n,m,q,\mathcal{X}}$ *problem is difficult, then the scheme in this paper is anonymous against opener in RO model.*

Proof Assume that the size of legitimate users is t, the adversary $\mathcal{A}$ and challenger $\mathcal{C}$ are all PPT algorithms. For two different users $i_0 \neq i_1 \in [t]$ given by $\mathcal{A}$,

the proof of property anonymity against opener is similar to that of anonymity against admitter, so we are not describe it in detail anymore. □

Lemma 3 *Suppose that the problem ring-*$\mathbf{SIS}^{\infty}_{n,m,q,1}$ *is difficult, then the sche-me in this paper is unforgeable in the RO model.*

Proof Suppose that there ia a PPT adversary $\mathcal{A}$ could forge a valid signature with a non-negligible probability ϵ, then there is a PPT algorithm $\mathcal{B}$ could break the security of Merkle hash tree or solve the problem ring-$\mathbf{SIS}^{\infty}_{n,m,q,1}$ with a non-negligible probability by invoking $\mathcal{A}$ as a black box.

If there is a negligible function $negl(\lambda)$, such that $\Pr\left[\mathbf{Exp}^{unforge}_{FDGS-MDO,\mathcal{A}}(\lambda) = 1\right] \leq negl(\lambda)$, then we say that the scheme is unforgeable. Given a random vector $\mathbf{A}$, the challenger computes the public parameter pp honestly, then invokes the algorithm of $\mathcal{A}$, runs the operations in the game $\mathbf{Exp}^{unforge}_{FDGS-MDO,\mathcal{A}}(\lambda)$, during this process, $\mathcal{B}$ responds the inquiries of $\mathcal{A}$ honestly. If the adversary $\mathcal{A}$ wins the game and outputs $\left(M^*, \Sigma^*, i^*, \Pi^*_{trace}, \mathbf{info}_\tau\right)$ finally, then there is a non-negligible function ϵ, such that $\Pr\left[\mathbf{Exp}^{unforge}_{FDGS-MDO,\mathcal{A}}(\lambda) = 1\right] \geq \epsilon$, and the algorithm $\mathcal{B}$ could operate as follows: Decompose the signature Σ^* into $(c^*_{1,1}, \mathbf{c}^*_2, \mathbf{c}^*_3, \mathbf{c}^*_4, \Pi^*_{sign})$, where $\Pi_{sign} = \left(\left\{\left(\mathbf{u}^*_j, \hat{\mathbf{u}}^*_j\right)\right\}_{j=1}^{k'}, \mathbf{ch}^*, \left\{rsp^*_j\right\}_{j=1}^{k'}\right)$, because the adversary $\mathcal{A}$ wins the game $\mathbf{Exp}^{unforge}_{FDGS-MDO,\mathcal{A}}(\lambda)$, so $\left\{rsp^*_j\right\}_{j=1}^{k'}$ is a legitimate respond to $\left\{\left(\mathbf{u}^*_j, \hat{\mathbf{u}}^*_j\right)\right\}_{j=1}^{k'}, \mathbf{ch}^*$. Let $\xi^* = \left(M^*, \left\{\left(\mathbf{u}^*_j, \hat{\mathbf{u}}^*_j\right)\right\}_{j=1}^{k'}, \mathbf{A}, \mathbf{u}_\tau, \{P_i\}_{i=1}^4, \mathbf{B}, c^*_{1,1}, \mathbf{c}^*_2, \mathbf{c}^*_3, \mathbf{c}^*_4\right)$, for

the successful probability to guess $H(\xi^*)$ is $(np)^{-k'}$, so the adversary uses the ξ^* to initiate queries to the oracle H with overwhelming probability, and ξ^* is the preimage of H with probability $\epsilon' = \epsilon - (np)^{-k'}$, let $t^* \in \{1, 2, \cdots, Q_H\}$ be the index of one inquiry, where Q_H is the number of inquiries that the adversary $\mathcal{A}$ made to the oracle H. The inputs of the hash queries from 1th to t^*th are all ξ^*, and $\mathcal{B}$ runs the operations of $\mathcal{A}$ for t^* times. And the inputs of other hash queries from $t^* + 1$th to Q_Hth are something else, $\mathcal{B}$ responds by independent values respectively. By the Forking lemma in (Brickell et al. 2000; Pointcheval and Stern 1999), the probability of $\mathcal{B}$ gets $\max(n, p) + 1$ different hash values $\mathbf{ch}_{t^*}^1, \cdots, \mathbf{ch}_{t^*}^{\max(n,p)+1} \in \{[n] \times \mathbb{Z}_p\}^{k'}$ to the same input ξ^* is non-negligible, and the pigeon hole principle tells us that there are at least two accept responds $(rsp_{t^*,1}, rsp_{t^*,2})$ with the same I and different ch, then what we could learn from the protocol that generates Π_{sign} is that we could extract a witness $\zeta' = \left(\mathbf{usk}_{i'}, \mathbf{upk}_{i'}, w'_\tau, \{\mathbf{r}'_i\}_{i=1}^4, \right)$, where $w'_\tau = \left(\mathbf{bin}(i'-1), \mathbf{w}'_{l,\tau}, \cdots, \mathbf{w}'_{1,\tau}\right) \in \{0,1\}^l \times (\{0,1\}^k)^l$, such that for $d = 1, 2$, $d' = 3, 4$, $j \in \{0, l-1\}$, we have

$$\begin{cases} \mathbf{u}_{j,\tau} = \begin{cases} h_{\mathbf{A}}(\mathbf{u}_{j+1,\tau}, \mathbf{w}_{j+1,\tau}), \ if\ i'_{j+1} = 0 \\ h_{\mathbf{A}}(\mathbf{w}_{j+1,\tau}, \mathbf{u}_{j+1,\tau}), \ if\ i'_{j+1} = 1 \end{cases} \\ \mathbf{A} \cdot \mathbf{usk}_{i'} = \mathbf{G} \cdot \mathbf{upk}_{i'} \\ \mathbf{c}^*_d = \left(c^*_{d,1}, \mathbf{c}^*_{d,2}\right) = \left(\mathbf{B} \cdot \mathbf{r}'_d, P_d \cdot \mathbf{r}'_d + \lfloor \frac{q}{2} \rceil \cdot \mathbf{upk}_{i'}\right) \\ \mathbf{c}^*_{d'} = \left(c^*_{d',1}, \mathbf{c}^*_{d',2}\right) = \left(\mathbf{B} \cdot \mathbf{r}'_{d'}, P_{d'} \cdot \mathbf{r}'_{d'} + \lfloor \frac{q}{2} \rceil \cdot \mathbf{c}^*_{1,2}\right) \end{cases} \tag{4}$$

We can learn from the correctness of the encryption scheme that $\mathbf{c}_1^*$ is the encryption to $\mathbf{upk}_{i'}$ and $\mathbf{c}_3^*$ is the encryption to $\mathbf{c}_{1,2}^*$. The algorithm **Judge** outputs 1 because of the fact that $\mathcal{A}$ wins the game, and what we can learn from the soundness of the protocol that generates Π_{trace} is that $\mathbf{c}_1^*$ is the encryption to $\mathbf{upk}_{i^*}$, then $\mathbf{upk}_{i'} = \mathbf{upk}_{i^*}$ with overwhelming probability. By the correctness of the Merkle hash tree, the user i^* is legitimate. $i^* \in HUL \setminus BUL$ indicates that the adversary $\mathcal{A}$ doesn't know $gsk_{i^*} = (\mathbf{bin}(i^* - 1), \mathbf{upk}_{i'}, \mathbf{usk}_{i^*})$. $\mathbf{usk}_{i^*}$ was chosen by $\mathcal{B}$ and $\mathbf{A} \cdot \mathbf{usk}_{i^*} = \mathbf{G} \cdot \mathbf{upk}_{i'}$, so we have $\Pr[\mathbf{usk}_{i^*} \neq \mathbf{usk}_{i'}] \geq \frac{1}{2}$. Let $\mathbf{z} = \mathbf{usk}_{i^*} - \mathbf{usk}_{i'}$, then $\mathbf{z} \neq \mathbf{0}$ and $\mathbf{Az} = 0 \mod q$, so, the algorithm $\mathcal{B}$ could solve the problem ring-$\mathbf{SIS}^\infty_{n,m,q,1}$ with non-negligible probability. □

Lemma 4 *Suppose that the ring-*$\mathbf{SIS}^\infty_{n,m,q,1}$ *problem is difficult, then the sche-me in this paper is traceable in RO model.*

Proof Given a negligible function $negl(\lambda)$, such that $\Pr[\mathbf{Exp}^{trace}_{FDGS-MDO,\mathcal{A}}(\lambda) = 1] \leq negl(\lambda)$, then we say that the scheme is traceable. In other words, If the adversary $\mathcal{A}$ wins the game $\mathbf{Exp}^{trace}_{FDGS-MDO,\mathcal{A}}(\lambda)$, the signature generated by $\mathcal{A}$ is legitimate and it was traced to a revoked user or a legitimate user without a valid proof Π_{trace} to it, and next, we will explain that the probability of the fact that the adversary $\mathcal{A}$ wins the game is negligible.

Let $(\mathbf{info}_\tau, M, \Sigma)$ be a forged information by the adversary $\mathcal{A}$ in the game $\mathbf{Exp}^{trace}_{FDGS-MDO,\mathcal{A}}(\lambda)$, then the challenger could extract the identity $(\mathbf{bin}(i-1), \Pi_{trace})$ by running the algorithm **Trace**. Decompose the signature Σ into $\left(c'_{1,1}, \mathbf{c}'_2, \mathbf{c}'_3, \mathbf{c}'_4, \Pi_{sign}\right)$, where $\Pi_{sign} = \left(\left\{\left(\mathbf{u}_j, \hat{\mathbf{u}}_j\right)\right\}_{j=1}^{k'}, \mathbf{ch}, \{rsp_j\}_{j=1}^{k'}\right)$. Since $(\mathbf{info}_\tau, M, \Sigma)$ is a legitimate signature, $\{rsp_j\}_{j=1}^{k'}$ are valid responds to $\left\{\left(\mathbf{u}_j, \hat{\mathbf{u}}_j\right)\right\}_{j=1}^{k'}$, $\mathbf{ch}$. Then we could extract a witness $\zeta' = \left(\mathbf{usk}_{i'}, \mathbf{upk}_{i'}, w'_\tau, \left\{\mathbf{r}'_i\right\}_{i=1}^4\right)$, which is similar to the property of unforgeability, where $w'_\tau = \left(\mathbf{bin}(i'-1), \mathbf{w}'_{l,\tau}, \cdots, \mathbf{w}'_{1,\tau}\right) \in \{0,1\}^l \times (\{0,1\}^k)^l$, such that for $d = 1, 2$, $d' = 3, 4$, $j \in \{0, l-1\}$, we have

$$\begin{cases} \mathbf{upk}_{i'} \neq 0 \\ \mathbf{u}_{j,\tau} = \begin{cases} h_{\mathbf{A}}(\mathbf{u}_{j+1,\tau}, \mathbf{w}_{j+1,\tau}), \ if\ i'_{i+1} = 0 \\ h_{\mathbf{A}}(\mathbf{w}_{j+1,\tau}, \mathbf{u}_{j+1,\tau}), \ if\ i'_{i+1} = 1 \end{cases} \\ \mathbf{A} \cdot \mathbf{usk}_{i'} = \mathbf{G} \cdot \mathbf{upk}_{i'} \\ \mathbf{c}'_d = \left(c'_{d,1}, \mathbf{c}'_{d,2}\right) = \left(\mathbf{B} \cdot \mathbf{r}'_d, P_d \cdot \mathbf{r}'_d + \lfloor \frac{q}{2} \rceil \cdot \mathbf{upk}_{i'}\right) \\ \mathbf{c}'_{d'} = \left(c'_{d',1}, \mathbf{c}'_{d',2}\right) = \left(\mathbf{B} \cdot \mathbf{r}'_{d'}, P_{d'} \cdot \mathbf{r}'_{d'} + \lfloor \frac{q}{2} \rceil \cdot \mathbf{c}'_{1,2}\right) \end{cases} \tag{5}$$

What we can learn from the correctness of the encryption scheme is that the ciphertext $\mathbf{c}'_1$ could be decrypted to $\mathbf{upk}_{i'}$, $\mathbf{c}'_3$ could be decrypted to $\mathbf{c}'_{1,2}$, and we can learn from the correctness of the algorithm **Trace** that $\mathbf{upk}_i$ is the plaintext obtained from the ciphertext $\mathbf{c}'_1$, so $\mathbf{upk}_i = \mathbf{upk}_{i'}$ with overwhelming probability, and the probability that a valid signature be traced to a revoked user is negligible. In fact, we can learn from the security of Merkle hash tree that the probability that the valid signature above be traced to a revoked user with a valid proof Π_{trace} is negligible. Because of the fact that the challenger has the legitimate witness to generate a valid proof Π_{trace}, and we can learn from the perfect completeness of the protocol that generates Π_{trace} that the algorithm **Judge** would accepts Π_{trace} with probability 1. In conclusion, the scheme in this paper is traceable. □

Lemma 5 *The scheme in this paper satisfies the property of tracing soundness in RO model.*

Proof Suppose that the information $(M, \Sigma, i_0, \Pi_{trace,i_0}, i_1, \Pi_{trace,i_1}, \mathbf{info}_\tau)$ is the output of the adversary $\mathcal{A}$ in the game $\mathbf{Exp}^{trace-sound}_{FDGS-MDO,\mathcal{A}}(\lambda)$, if the game outputs 1 finally, i.e. $\mathbf{Judge}(gpk, \mathbf{upk}_{i_b}, \mathbf{info}_\tau, \Pi_{trace}, M, \Sigma) = 1$, $i_0 \neq i_1 \neq \perp$, $\mathbf{Verify}(gpk, \mathbf{info}_\tau, M, \Sigma) = 1$, then we say that $\mathcal{A}$ wins.

Given Π_{trace} with $\Pi_{trace} = \left(\{(\mathbf{u}_j, \hat{\mathbf{u}}_j)\}_{j=1}^{k'}, \mathbf{ch}, \{rsp_j\}_{j=1}^{k'}\right)$, the fact that the algorithm **Judge** outputs 1 indicates that $\{rsp_j\}_{j=1}^{k'}$ are legitimate responds to $\{(\mathbf{u}_j, \hat{\mathbf{u}}_j)\}_{j=1}^{k'}, \mathbf{ch}$. For $b = 0, 1$, $j = 1, 3$, it is similarly to the property of unforgeability, we could extract $\mathbf{S}_{1,b}, \tilde{\mathbf{S}}_{3,b}, E_{j,b}, \mathbf{y}_{j,b}$, such that

$$\begin{aligned}
\|\mathbf{S}_{1,b}\|_\infty, \|\tilde{\mathbf{S}}_{3,b}\|_\infty \le \beta, |E_{j,b}| \le \beta, \|\mathbf{y}_{j,b}\|_\infty \le \left\lceil \frac{q}{5} \right\rceil \\
\mathbf{S}_{1,b}^\top \cdot \mathbf{B} + E_{1,b} = P_{1,b} \mod q \\
\tilde{\mathbf{S}}_{3,b}^\top \cdot \mathbf{B} + E_{3,b} = P_{3,b} \mod q \\
\mathbf{c}_{3,2} - \tilde{\mathbf{S}}_{3,b}^\top \cdot c_{3,1} = \mathbf{y}_{3,b} + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathbf{c}_{1,2} \mod q \\
\mathbf{c}_{1,2} - \mathbf{S}_{1,b}^\top \cdot c_{1,1} = \mathbf{y}_{1,b} + \left\lfloor \frac{q}{2} \right\rfloor \cdot \mathbf{upk}_{i_b} \mod q
\end{aligned}$$

then we have

$$\left(\mathbf{S}_{1,0}^\top - \mathbf{S}_{1,1}^\top\right) \cdot c_{1,1} = (\mathbf{y}_{1,1} - \mathbf{y}_{1,0}) + \left\lfloor \frac{q}{2} \right\rfloor \cdot (\mathbf{upk}_{i_1} - \mathbf{upk}_{i_0}) \mod q$$

Suppose that $\mathbf{upk}_{i_1} \neq \mathbf{upk}_{i_0}$, so $\|\lfloor \frac{q}{2} \rfloor \cdot (\mathbf{upk}_{i_1} - \mathbf{upk}_{i_0})\|_\infty = \lfloor \frac{q}{2} \rfloor$, $\|\mathbf{y}_{1,1} - \mathbf{y}_{1,0}\|_\infty \le 2 \cdot \lceil \frac{q}{5} \rceil$, and

$$\|(\mathbf{y}_{1,1} - \mathbf{y}_{1,0}) + \lfloor \frac{q}{2} \rfloor \cdot (\mathbf{upk}_{i_1} - \mathbf{upk}_{i_0})\|_\infty > 0$$

then $\mathbf{S}_{1,0}^\top \neq \mathbf{S}_{1,1}^\top$, we obtained two different solutions of the function $\mathbf{S}_1^\top \cdot \mathbf{B} + E_1 = P_1 \mod q$, which is contradictory to the fact that there is at most one solution to the ring-$\mathbf{LWE}_{n,m,q,\mathcal{X}}$ sample $(\mathbf{B}, P_1)$. So, $\mathbf{upk}_{i_1} = \mathbf{upk}_{i_0}$ with overwhelming probability. Similarly, if there are two different strings $\mathbf{c}_{1,2}$ and $\mathbf{c}'_{1,2}$ w.r.t one ciphertext $\mathbf{c}_3$, then $\mathbf{c}_{1,2} = \mathbf{c}'_{1,2}$ is also true with overwhelming probability. In other words, the probability of the fact that $\mathcal{A}$ wins is negligible, so the scheme in this paper satisfies the property of tracing soundness. □

The improved zero-knowledge protocol of knowledge

Details of the protocol

Suppose that the size of the legitimate members in the group is $t \ge 1$ at time τ, for $d = 1, 2$, $d' = 3, 4$, $i \in [t]$, $\forall j \in [l-1]$, the underlying zero-knowledge protocol is used to prove the following relationships by utilizing the extending and permuting techniques (Stern 1996; Ling et al. 2017).

$$\begin{cases}
\mathbf{upk}_i \neq 0 \\
\mathbf{u}_j = \begin{cases} h_{\mathbf{A}}(\mathbf{u}_{j+1}, \mathbf{w}_{j+1}), \ if\ i_{j+1} = 0 \\ h_{\mathbf{A}}(\mathbf{w}_{j+1}, \mathbf{u}_{j+1}), \ if\ i_{j+1} = 1 \end{cases} \quad (\star) \\
\mathbf{upk}_i = \mathbf{bin}(\mathbf{A} \cdot \mathbf{usk}_i) \\
\mathbf{c}_d = (c_{d,1}, \mathbf{c}_{d,2}) = (\mathbf{B} \cdot \mathbf{r}_d, P_d \cdot \mathbf{r}_d + \lfloor \frac{q}{2} \rceil \cdot \mathbf{upk}_i) \\
\mathbf{c}_{d'} = (c_{d',1}, \mathbf{c}_{d',2}) = (\mathbf{B} \cdot \mathbf{r}_{d'}, P_{d'} \cdot \mathbf{r}_{d'} + \lfloor \frac{q}{2} \rceil \cdot \mathbf{c}_{1,2})
\end{cases} \tag{6}$$

Given a bit b, a vector $\mathbf{a}$, let $\mathbf{ext}(b, \mathbf{a}) = (\bar{b} \cdot \mathbf{a}, b \cdot \mathbf{a})^\top$, $\mathbf{ext}_2(b) = (\bar{b}, b)^\top$. Given bit b' and vector $\mathbf{a}'$, we can get similar results $\mathbf{ext}(b', \mathbf{a}') = (\bar{b}' \cdot \mathbf{a}', b' \cdot \mathbf{a}')^\top$, $\mathbf{ext}_2(b') = (\bar{b}', b')^\top$. then we have the following equivalence relationship:

$$\begin{aligned}
(\star) &\Leftrightarrow \bar{i}_{j+1} \cdot h_{\mathbf{A}}(\mathbf{u}_{j+1}, \mathbf{w}_{j+1}) + i_{j+1} \cdot h_{\mathbf{A}}(\mathbf{w}_{j+1}, \mathbf{u}_{j+1}) = \mathbf{u}_j \\
&\Leftrightarrow \bar{i}_{j+1} \cdot (\mathbf{A}_0 \mathbf{u}_{j+1} + \mathbf{A}_1 \mathbf{w}_{j+1}) + i_{j+1} \cdot (\mathbf{A}_0 \mathbf{w}_{j+1} + \mathbf{A}_1 \mathbf{u}_{j+1}) = \mathbf{G}\mathbf{u}_j \mod q \\
&\Leftrightarrow \mathbf{A} \cdot \begin{pmatrix} \bar{i}_{j+1} \cdot \mathbf{u}_{j+1} \\ i_{j+1} \cdot \mathbf{u}_{j+1} \end{pmatrix} + \mathbf{A} \cdot \begin{pmatrix} i_{j+1} \cdot \mathbf{w}_{j+1} \\ \bar{i}_{j+1} \cdot \mathbf{w}_{j+1} \end{pmatrix} = \mathbf{G} \cdot \mathbf{u}_j \mod q \\
&\Leftrightarrow \mathbf{A} \cdot \mathbf{ext}(i_{j+1}, \mathbf{u}_{j+1}) + \mathbf{A} \cdot \mathbf{ext}(\bar{i}_{j+1}, \mathbf{w}_{j+1}) = \mathbf{G} \cdot \mathbf{u}_j \mod q
\end{aligned}$$

Then for $d = 1, 2$, $d' = 3, 4$, $i \in [t]$, $\mathbf{bin}(i-1) = (i_1, \cdots, i_l)$, the Eq. (2) is equal to the following form

$$\begin{cases}
\mathbf{A} \cdot \mathbf{ext}(i_1, \mathbf{u}_1) + \mathbf{A} \cdot \mathbf{ext}(\bar{i}_1, \mathbf{w}_1) - \mathbf{G} \cdot \mathbf{u} = 0 \mod q \\
\mathbf{A} \cdot \mathbf{ext}(i_2, \mathbf{u}_2) + \mathbf{A} \cdot \mathbf{ext}(\bar{i}_2, \mathbf{w}_2) - \mathbf{G} \cdot \mathbf{u}_1 = 0 \mod q \\
\qquad \cdots \\
\mathbf{A} \cdot \mathbf{ext}(i_l, \mathbf{upk}_i) + \mathbf{A} \cdot \mathbf{ext}(\bar{i}_l, \mathbf{w}_l) - \mathbf{G} \cdot \mathbf{u}_{l-1} = 0 \mod q \\
\mathbf{A} \cdot \mathbf{usk}_i - \mathbf{G} \cdot \mathbf{upk}_i = 0 \mod q \\
c_{d,1} = \mathbf{B} \cdot \mathbf{r}_d \mod q \\
\mathbf{c}_{d,2} = P_d \cdot \mathbf{r}_d + \lfloor \frac{q}{2} \rceil \cdot \mathbf{upk}_i \mod q \\
c_{d',3} = \mathbf{B} \cdot \mathbf{r}_{d'} \mod q \\
\mathbf{c}_{d',2} = P_{d'} \cdot \mathbf{r}_{d'} + \lfloor \frac{q}{2} \rceil \cdot \mathbf{c}_{1,2} \mod q
\end{cases} \tag{7}$$

Let $\mathbf{B}_n^{2n}$ be the set of strings with length $2n$, where the Hamming weight of each string is n, to illustrate the fact that the user's public key $\mathbf{upk}_i \neq 0^k$, we pad $\mathbf{upk}_i$ with a random string with length $k-1$ to obtain a new string $\mathbf{upk}_i^*$, such that $\mathbf{upk}_i^* \in \mathbf{B}_k^{2k-1}$, then for any permutation $\pi_{\mathbf{upk}_i} \in \mathcal{S}_{2k-1}$, we have

$$\mathbf{upk}_i \neq 0^k \Leftrightarrow \mathbf{upk}_i^* \in \mathbf{B}_k^{2k-1} \Leftrightarrow \pi_{\mathbf{upk}_i}(\mathbf{upk}_i^*) \in \mathbf{B}_k^{2k-1}$$

We make similar operations for $\mathbf{c}_{1,2}$ to obtain $\mathbf{c}_{1,2}^* \in \mathbf{B}_k^{2k-1}$, for each $\mathbf{usk}_i$ to obtain $\mathbf{usk}_i^* \in \mathbf{B}_m^{2m}$, for any $\pi_{\mathbf{usk}_i} \in \mathcal{S}_{2m}$, we have $\mathbf{usk}_i^* \in \mathbf{B}_m^{2m} \Leftrightarrow \pi_{\mathbf{usk}_i}(\mathbf{usk}_i^*) \in \mathbf{B}_m^{2m}$. Similarly, extend the vectors $\mathbf{u}_1, \cdots, \mathbf{u}_{l-1}, \mathbf{w}_1, \cdots, \mathbf{w}_l, \mathbf{r}_1, \cdots, \mathbf{r}_4$ to obtain $\mathbf{u}_1^* \cdots, \mathbf{u}_{l-1}^*$, $\mathbf{w}_1^* \cdots, \mathbf{w}_l^* \in \mathbf{B}_k^{2k}$, $\mathbf{r}_1^*, \cdots, \mathbf{r}_4^* \in \mathbf{B}_k^{2k}$. And then let $\hat{\mathbf{u}}_1 = \mathbf{ext}(i_1, \mathbf{u}_1^*), \cdots, \hat{\mathbf{u}}_{l-1} = \mathbf{ext}(i_{l-1}, \mathbf{u}_{l-1}^*) \in \{0,1\}^{4k}$, $\hat{\mathbf{upk}}_i = \mathbf{ext}(i_l, \mathbf{upk}_i^*) \in \{0,1\}^{4k-2}$, $\hat{\mathbf{w}}_1 = \mathbf{ext}(\bar{i}_1, \mathbf{w}_1^*), \cdots, \hat{\mathbf{w}}_l = \mathbf{ext}(\bar{i}_l, \mathbf{w}_l^*) \in \{0,1\}^{4k}$.

Given $\mathbf{upk}_i = (upk_{i1}, \cdots, upk_{ik})$, for any $j \in [k]$, let $\mathbf{upk}'_{ij} = \mathbf{ext}_2(upk_{ij})$. For any $b \in \{0,1\}$, $\mathbf{t} = (t_0, t_1) \in \mathbb{Z}^2$, let $T_b(\mathbf{t}) = (t_b, t_{\bar{b}})$. Then for any $b_j \in \{0,1\}$, we have $\mathbf{upk}'_{ij} = \mathbf{ext}_2(upk_{ij}) \Leftrightarrow T_{b_j}(\mathbf{upk}'_{ij}) = \mathbf{ext}_2(upk_{ij} \oplus b_j)$. Because b_j is chosen randomly, so the operations above are equal to carry out a one-time pad to the user's upk_{ij} by b_j to hide it perfectly. And for $\mathbf{c}_{1,2}$ and $\mathbf{t}' = (t'_0, t'_1) \in \mathbb{Z}^2$, we give similar operations.

Let $r \in \{2k-1, 2k\}$, $b \in \{0,1\}$, $\pi \in \mathcal{S}_r$, $\mathbf{t} = (t_0, t_1)^T \in \mathbb{Z}^{2r}$, $\mathbf{t}' = (t'_0, t'_1) \in \mathbb{Z}^2$, we define the permutation $F_{b,\pi}(\mathbf{t}) = (\pi(t_b), \pi(t_{\bar{b}}))$, $F_{b,\pi}(\mathbf{t}') = (\pi(t'_b), \pi(t'_{\bar{b}}))$. Then

for all $b_1, \cdots, b_l \in \{0,1\}$, $\phi_{\mathbf{u},1}, \cdots, \phi_{\mathbf{u},l-1}, \phi_{\mathbf{w},1}, \cdots, \phi_{\mathbf{w},l} \in \mathcal{S}_{2k}$, $\pi_{\mathbf{upk}_i}, \pi_{\mathbf{c}_1} \in \mathcal{S}_{2k-1}$, the following relationship is true,

$$\begin{cases} \forall j \in [\,l-1\,], \hat{\mathbf{u}}_j = \mathbf{ext}(i_j, \mathbf{u}_j^*) \Leftrightarrow F_{b_j,\phi_{\mathbf{u},j}}(\hat{\mathbf{u}}_j) = \mathbf{ext}(i_j \oplus b_j, \phi_{\mathbf{u},j}(\mathbf{u}_j^*)) \\ \forall j \in [\,l\,], \hat{\mathbf{w}}_j = \mathbf{ext}(i_j, \mathbf{w}_j^*) \Leftrightarrow F_{b_j,\phi_{\mathbf{w},j}}(\hat{\mathbf{w}}_j) = \mathbf{ext}(i_j \oplus b_j, \phi_{\mathbf{w},j}(\mathbf{w}_j^*)) \\ \hat{\mathbf{upk}}_i = \mathbf{ext}(i_l, \mathbf{upk}_i^*) \Leftrightarrow F_{b_l,\pi_{\mathbf{upk}_i}}(\hat{\mathbf{upk}}_i) = \mathbf{ext}(i_l \oplus b_l, \pi_{\mathbf{upk}_i}(\mathbf{upk}_i^*)) \\ \hat{\mathbf{c}_{1,2}} = \mathbf{ext}(i_l, \mathbf{c}_{1,2}^*) \Leftrightarrow F_{b_l,\pi_{\mathbf{c}_{1,2}}}(\hat{\mathbf{c}_{1,2}}) = \mathbf{ext}(i_l \oplus b_l, \pi_{\mathbf{c}_{1,2}}(\mathbf{c}_{1,2}^*)) \end{cases} \tag{8}$$

Let

$$\begin{aligned} \mathbf{z} = (&\mathbf{u}_1^* \| \hat{\mathbf{u}}_1 \| \hat{\mathbf{w}}_1 \| \cdots \| \mathbf{u}_{l-1}^* \| \hat{\mathbf{u}}_{l-1} \| \hat{\mathbf{w}}_{l-1} \| \mathbf{upk}_i^* \| \hat{\mathbf{upk}}_i \| \\ &\mathbf{c}_{1,2}^* \| \hat{\mathbf{c}_{1,2}} \| \hat{\mathbf{w}}_l \| \mathbf{usk}_i^* \| \mathbf{r}_1^* \| \cdots \| \mathbf{r}_4^* \| \mathbf{upk}_{i1}' \| \cdots \| \mathbf{upk}_{ik}') \end{aligned}$$

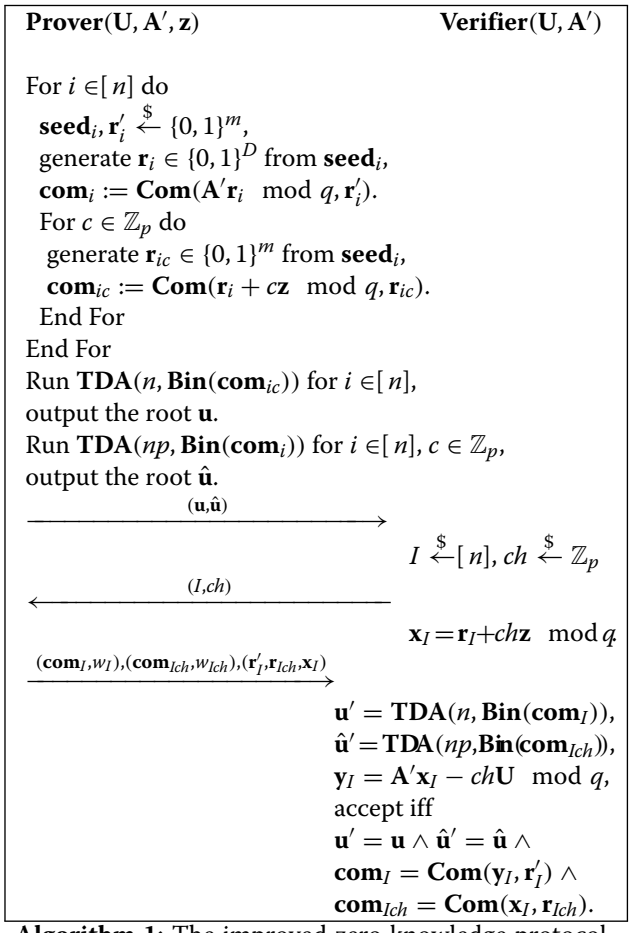
then $\mathbf{z} \in \{0,1\}^{10kl+2m+16k-6}$, the equation (4) can be unified into one equation $\mathbf{A}'\mathbf{z} = \mathbf{U} \mod q$, where $\mathbf{A}', \mathbf{U}$ could be obtained from the public parameters. Let **VALID** be the set of vectors in $\{0,1\}^{10kl+2m+16k-6}$ that satisfy the relationship above, let $\bar{\mathcal{S}} = \mathcal{S}_{2k}^{2l-1} \times \mathcal{S}_{2k-1}^2 \times \mathcal{S}_{2m} \times \mathcal{S}_{2l}^4 \times \{0,1\}^l$ for any

$$\begin{aligned} \eta = ((&\phi_{\mathbf{u},1}, \cdots, \phi_{\mathbf{u},l-1}, \phi_{\mathbf{w},1}, \cdots, \phi_{\mathbf{w},l}), \pi_{\mathbf{upk}_i}, \\ &\pi_{\mathbf{c}_{1,2}}, \pi_{\mathbf{usk}_i}, (\pi_{r,1}, \cdots, \pi_{r,4}), (b_1, \cdots, b_l)) \in \bar{\mathcal{S}} \end{aligned}$$

let Γ_η be the permutation for strings in $\{0,1\}^{10kl+2m+16k-6}$, then we have

$$\mathbf{z} \in \mathbf{VALID} \Leftrightarrow \Gamma_\eta(\mathbf{z}) \in \mathbf{VALID}$$

After that, we could utilize our protocol and the equal relationship above to proof that $\mathbf{z} \in \mathbf{VALID}$, and $\mathbf{A}'\mathbf{z} = \mathbf{U} \mod q$. Let $D = 10kl + 2m + 16k - 6$, the protocol is presented in Algorithm 1, where the commitment $Com : \{0,1\}^* \times \{0,1\}^m \to \mathbb{Z}_q^n$ is a string commitment scheme with properties of statistical hiding and computational binding (Kawachi et al. 2008).



Algorithm 1: The improved zero knowledge protocol of knowledge

Security analysis of the protocol

Theorem 4 *Suppose that the commitment scheme used in this paper satisfies statistical hiding and computing binding, then our new zero knowledge protocol satisfies completeness,* $(\max(n,p)+1)$*-special soundness and special honest-verifier zero knowledge.*

Proof Denote $rsp = ((\mathbf{com}_I, w_I), (\mathbf{com}_{Ich}, w_{Ich}), (\mathbf{r}'_I, \mathbf{r}_{Ich}, \mathbf{x}_I))$, we prove completeness, $(\max(n,p)+1)$-special soundness and special honest-verifier zero knowledge separately:

Completeness: Suppose that the prover and verifier have run each step of the protocol honestly, then $\mathbf{u}' = \mathbf{u} \wedge \hat{\mathbf{u}}' = \hat{\mathbf{u}}$ is true with overwhelming probability by the definition of **TDA**, and we have

$$\mathbf{y}_I = \mathbf{A}'\mathbf{x}_I - ch\mathbf{U} \mod q = \mathbf{A}'\mathbf{r}_I + ch\mathbf{A}'\mathbf{z} - ch\mathbf{U} \mod q.$$

So if $\mathbf{z}$ is a solution to the instance $(\mathbf{U}, \mathbf{A}')$, then $\mathbf{U} = \mathbf{A}'\mathbf{z} \mod q$, which means that $\mathbf{y} = \mathbf{A}'\mathbf{r}_I$, and the completeness of the protocol follow from the binding of the commitment scheme.

$(\max(n,p)+1)$-**special soundness**: If there are $(\max(n,p)+1)$ valid transcripts, the pigeon hole principle tells us that there are at least two accept transcripts with the same I and different ch. Suppose $\big((\mathbf{u}, \hat{\mathbf{u}}), (I, ch), \big((\mathbf{com}_I, w_I), (\mathbf{com}_{Ich}, w_{Ich}), \big(\mathbf{r}'_I, \mathbf{r}_{Ich}, \mathbf{x}_I\big)\big)\big)$ and $((\mathbf{u}, \hat{\mathbf{u}}), (I, ch'), ((\mathbf{com}_I, w_I), (\mathbf{com}_{Ich'}, w_{Ich'}), (\mathbf{r}'_I, \mathbf{r}_{Ich'}, \mathbf{x}'_I)))$ are two valid transcripts with $ch \neq ch'$, one can efficiently extract a collision of the hash function $h_{\mathbf{A}} \in \mathcal{H}$, a witness $\mathbf{z}$ such that $\mathbf{U} = \mathbf{A}'\mathbf{z}$ by using the binding of the commitment scheme.

Suppose that $\big((\mathbf{aux}_I, \mathbf{com}_I), (I, ch), \big(\mathbf{r}'_I, \mathbf{r}_{Ich}, \mathbf{x}_I\big)\big)$ and $((\mathbf{aux}_I, \mathbf{com}_I), (I, ch'), (\mathbf{r}'_I, \mathbf{r}_{Ich'}, \mathbf{x}'_I))$ are two valid transcripts that are accepted by verifier. Let $\mathbf{y}_I = \mathbf{A}'\mathbf{x}_I - ch\mathbf{U} \mod q$ and $\mathbf{y}'_I = \mathbf{A}'\mathbf{x}'_I - ch'\mathbf{U} \mod q$, then we have $\mathbf{com}_I = \mathbf{Com}\big(\mathbf{y}_I, \mathbf{r}'_I\big) = \mathbf{Com}\big(\mathbf{y}'_I, \mathbf{r}'_I\big)$, so the binding of the commitment implies that $\mathbf{y}_I = \mathbf{y}'_I$, i.e. $\mathbf{A}'(\mathbf{x}_I - \mathbf{x}'_I) = (ch - ch')\mathbf{U}$.

In addition, $\mathbf{com}_{Ich} = \mathbf{Com}(\mathbf{r}_I + ch\mathbf{z} \mod q, \mathbf{r}_{Ich}) = \mathbf{Com}(\mathbf{x}_I, \mathbf{r}_{Ich})$ and $\mathbf{com}_{Ich'} = \mathbf{Com}(\mathbf{r}_I + ch'\mathbf{z} \mod q, \mathbf{r}_{Ich'})$

$= \mathbf{Com}(\mathbf{x}'_I, \mathbf{r}_{Ich'})$, so $\mathbf{x}_I = \mathbf{r}_I + ch\mathbf{z} \mod q$ and $\mathbf{x}'_I = \mathbf{r}_I + ch'\mathbf{z} \mod q$ by the binding of the commitment.

$$\left.\begin{aligned} \mathbf{y}_I &= \mathbf{y}'_I \\ \mathbf{A}'\mathbf{x}_I - ch\mathbf{U} \mod q &= \mathbf{y}_I \\ \mathbf{A}'\mathbf{x}'_I - ch'\mathbf{U} \mod q &= \mathbf{y}'_I \\ \mathbf{x}_I = \mathbf{r}_I + ch\mathbf{z} \mod q & \\ \mathbf{x}'_I = \mathbf{r}_I + ch'\mathbf{z} \mod q & \end{aligned}\right\} \Longrightarrow \mathbf{x}_I - \mathbf{x}'_I = (ch - ch')\mathbf{z}$$

Then one can compute $\mathbf{z}$ efficiently as a solution of the instance $(\mathbf{U}, \mathbf{A}')$.

Special honest-verifier zero knowledge: In this proof, we construct a PPT simulator $\mathcal{S}$ with inputs $(\mathbf{U}, \mathbf{A}')$, $\{\mathbf{seed}_i\}_{i\in[n]}$ and (I, ch), it interacts with a (maybe dishonest) verifier and does the following things:

1. Sample $\mathbf{r}'_I \xleftarrow{\$} \{0,1\}^m$, and compute $\mathbf{r}_I$, $\mathbf{r}_{Ich}$ from $\mathbf{seed}_I$.
2. Compute $\mathbf{com}_I = \mathbf{Com}(\mathbf{A}'\mathbf{r}_I \mod q, \mathbf{r}'_I)$ honestly, commit to random dummy values to calculate the commitments $\mathbf{com}_{i\neq I}$.
3. Compute a vector $\mathbf{z}'$ by Gaussian elimination such that $\mathbf{U} = \mathbf{A}'\mathbf{z}' \mod q$.
4. Compute $\mathbf{x}'_I = \mathbf{r}_I + ch\mathbf{z}' \mod q$, $\mathbf{com}_{Ich} = \mathbf{Com}(\mathbf{x}'_I, \mathbf{r}_{Ich})$, and commit to random dummy values to calculate the commitments $\mathbf{com}_{ic}$ for all $i \neq I$ and $c \neq ch$.
5. Run $\mathbf{TDA}(n, \mathbf{Bin}(\mathbf{com}_i))$ for $i \in [n]$, $\mathbf{TDA}(np, \mathbf{Bin}(\mathbf{com}_{ic}))$ for $i \in [n]$, $c \in \mathbb{Z}_p$, output the root $\mathbf{u}'$ and $\hat{\mathbf{u}}'$ respectively.
6. Output the transcript $((\mathbf{u}', \hat{\mathbf{u}}'), (I, ch), ((\mathbf{com}_I, w_I), (\mathbf{com}_{Ich}, w_{Ich}), (\mathbf{r}'_I, \mathbf{r}_{Ich}, \mathbf{x}'_I)))$.

It is clear that $(\mathbf{r}'_I, \mathbf{r}_{Ich}, \mathbf{x}'_I)$ and the corresponding real transcript are both uniformly distributed in $\{0,1\}^{2\lambda} \times \{0,1\}^D$ and hence follow the same distribution. $(\mathbf{com}_I, \mathbf{com}_{Ich})$ and the corresponding real transcript are statistical indistinguishable by the hiding property of the commitment. By the definition of the collision resistant hash function, both (w_I, w_{Ich}) and the corresponding real transcript are indistinguishable from uniform distribution, so (w_I, w_{Ich}) and the corresponding real transcript are indistinguishable. Because the commitments $\mathbf{com}_i$, $\mathbf{com}_{ic}$ for all $i \neq I$, $c \neq ch$ are never opened, $(\mathbf{u}', \hat{\mathbf{u}}')$ also follows from the hiding property of the commitment and the definition of the hash function. So, the transcript outputted by $\mathcal{S}$ and the real transcript of the protocol are computing indistinguishable. □

Conclusion

In this paper, we give a new ring-based fully dynamic group signature scheme with message-dependent opening. The efficiency of it is improved by an improved underlying zero knowledge proof of knowledge that has smaller soundness error than Stern-like protocol. This modification helps to bring down the communication complexity of the underlying zero knowledge protocol and hence the computational/space complexity of the group signature scheme. In addition, we add another participant - an admitter to our scheme to constrain the power of trace manager. The admitter could generate tokens with respect to messages by using its secret key such that the trace manager can only open signatures of messages specified by the admitter.

Acknowledgements
Not applicable.

Authors' contributions
The first author conceived the idea of the study and wrote the paper; all authors discussed the results and revised the final manuscript. Both authors read and approved the final manuscript.

Funding
This work was supported by the National Natural Science Foundation of China (Grant No.61932019, No.61772521, No.61772522) and the Key Research Program of Frontier Sciences, CAS (Grant No.QYZDB-SSW-SYS035).

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Haidian District, Beijing, China. [2] School of Cyber Security, University of Chinese Academy of Sciences, Huairou District, China. [3] State Key Laboratory of Cryptology, P.O. Box 515, Xicheng District, Beijing, China.

Received: 26 October 2020 Accepted: 1 February 2021
Published online: 03 May 2021

References

Beullens W (2020) Sigma protocols for mq, pkp and sis, and fishy signature schemes. In: Canteaut A, Ishai Y (eds). Proceedings of Conference EUROCRYPT: 10-14 May 2020. Springer, Zagreb. pp 183–211

Bootle J, Cerulli A, Chaidos P, Ghadafi E, Groth J (2016) Foundations of fully dynamic group signatures. In: Manulis M, Sadeghi A-R, Schneider S (eds). Proceedings of Conference ACNS: 19-22 June 2016. Springer, Guildford. pp 117–136

Brickell E, Pointcheval D, Vaudenay S, Yung M (2000) Design validations for discrete logarithm based signature schemes. In: Imai H, Zheng Y (eds). Proceedings of Conference PKC: 18-20 January 2000. Springer, Melbourne. pp 276–292

Canetti R, Halevi S, Katz J (2004) Chosen-ciphertext security from identity-based encryption. In: Cachin C, Camenisch J (eds). Proceedings of Conference EUROCRYPT: 2-6 May 2004. Springer, Interlaken. pp 207–222

Chaum D, van Heyst E (1991) Group signatures. In: Davies DW (ed). Proceedings of Conference EUROCRYPT: 8-11 April 1991. Springer, Brighton. pp 257–265

Gordon SD, Katz J, Vaikuntanathan V (2010) A group signature scheme from lattice assumptions. In: Abe M (ed). Proceedings of Conference ASIACRYPT: 5-9 December 2010. Springer, Singapore. pp 395–412

Hazay C, Lindell Y (2010) Sigma protocols and efficient zero-knowledge. In: Efficient Secure Two-Party Protocols. Information Security and Cryptography. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-14303-8_6

Katsumata S, Yamada S (2019) Group signatures without nizk: from lattice in the standard model. In: Ishai Y, Rijmen V (eds). Proceedings of Conference EUROCRYPT: 19-23 May 2019. Springer, Darmstadt. pp 312–344

Kawachi A, Tanaka K, Xagawa K (2008) Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In: Pieprzyk J (ed). Proceedings of Conference ASIACRYPT: 7-11 December 2008. Springer, Singapore. pp 372–389
Laguillaumie F, Langlois A, Libert B, Stehlé D (2013) Lattice-based group signatures with logarithmic signature size. In: Sako K, Sarkar P (eds). Proceedings of Conference ASIACRYPT: 1-5 December 2013. Springer, Bengaluru. pp 41–61
Langlois A, Ling S, Nguyen K, Wang H (2014) Lattice-based group signature scheme with verifier-local revocation. In: Krawczyk H (ed). Proceedings of Conference PKC: 26-28 March 2014. Springer, Buenos. pp 345–361
Libert B, Joye M (2014) Group signatures with message-dependent opening in the standard model. In: Benaloh J (ed). Proceedings of Conference CT-RSA: 25-28 February 2014. Springer, San Francisco. pp 286–306
Libert B, Ling S, Nguyen K, Wang H (2016) Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without trapdoors. In: Fischlin M, Coron J-S (eds). Proceedings of Conference EUROCRYPT: 8-12 May 2016. Springer, Vienna. pp 1–31
Libert B, Mouhartem F, Nguyen K (2016) A lattice-based group signature scheme with message-dependent opening. In: Manulis M, Sadeghi A-R, Schneider S (eds). Proceedings of Conference ACNS: 19-22 June 2016. Springer, Guildford. pp 137–155
Ling S, Nguyen K, Wang HX (2015) Group signatures from lattices: simpler, tighter, shorter, ring-based. In: Katz J (ed). Proceedings of Conference PKC: 30 March-1 April 2015. Springer, Gaithersburg. pp 427–449
Ling S, Nguyen K, Wang H, Xu Y (2017) Lattice-based group signatures: achieving full dynamicity with ease. In: Gollmann D, Miyaji A, Kikuchi H (eds). Proceedings of Conference ACNS: 10-12 July 2017. Springer, Kanazawa. pp 293–312
Lyubashevsky V (2008) Lattice-based identification schemes secure under active attacks. In: Cramer R (ed). Proceedings of Conference PKC: 9-12 March 2008. Springer, Barcelona. pp 162–179
Lyubashevsky V (2012) Lattice signatures without trapdoors. In: Pointcheval D, Johansson T (eds). Proceedings of Conference EUROCRYPT: 15-19 April 2012. Springer, Cambridge. pp 738–755
Lyubashevsky V, Micciancio D (2006) Generalized compact knapsacks are collision resistant. In: Bugliesi M, Preneel B, Sassone V, Wegener I (eds). Proceedings of Conference ICALP: 10-14 July 2006. Springer, Venice. pp 144–155
Lyubashevsky V, Peikert C, Regev O (2010) On ideal lattices and learning with errors over rings. In: Gilbert H (ed). Proceedings of Conference EUROCRYPT: 30 May-3 June 2010. Riviera, Springer. pp 1–23
Lyubashevsky V, Peikert C, Regev O (2013) A toolkit for ring-lwe cryptography. In: Johansson T, Q.Nguyen P (eds). Proceedings of Conference EUROCRYPT: 26-30 May 2013. Springer, Athens. pp 35–54
Naor M, Yung M (1990) Public-key cryptosystems provably secure against chosen ciphertext attacks. In: Proceedings of the ACM Conference STOC: 1990. ACM DL, Baltimore. pp 427–437
Ohara K, Sakai Y, Emura K, Hanaoka G (2013) A group signature scheme with unbounded message-dependent opening. In: Proceedings of the ACM Conference AsiaCCS: 2013. ACM DL, Hangzhou. pp 517–522
Peikert C (2016) A decade of lattice cryptography. Found Trends Theor Comput Sci 10:283–424
Peikert C, Rosen A (2006) Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: Halevi S, Rabin T (eds). Proceedings of Conference TCC: 4-7 March 2006. Springer, New York. pp 145–166
Peikert C, Rosen A (2007) Lattices that admit logarithmic worst-case to average-case connection factors. In: Proceedings of the ACM Conference STOC: 11-13 June 2007. ACM DL, San Diego. pp 478–487
Pointcheval D, Stern J (1999) Security arguments for digital signatures and blind signatures. J Cryptol 13(3):361–396
Regev O (2009) On lattices, learning with errors, random linear codes, and cryptography. J ACM 56:1–40
Sakai Y, Emura K, Hanaoka G, Kawai Y, Matsuda T, Omote K (2012) Group signatures with message-dependent opening. In: Abdalla M, Lange T (eds). Proceedings of Conference Pairing: 16-18 May 2012. Springer, Cologne. pp 270–294
Stern J (1996) A new paradigm for public key identification. IEEE Trans Inf Theory 42(6):1757–1768
Sun Y, Liu Y (2020) A lattice-based fully dynamic group signature scheme without nizk. In: Proceedings of Conference INSCRYPT: 11-14 December 2020. Springer, Guangzhou
Sun Y, Liu Y, Wu B (2019) An efficient full dynamic group signature scheme over ring. Cybersecurity 2(21). https://doi.org/10.1186/s42400-019-0037-8

Publisher's Note