This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087808, IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Identity-based Linkable Ring Signatures from Lattices

HUY QUOC LE[1,2], (Student Member, IEEE), BAY VO[3], DUNG HOANG DUONG[1], WILLY SUSILO[1], (Fellow, IEEE), NGOC T. LE[1], KAZUHIDE FUKUSHIMA[4], SHINSAKU KIYOMOTO[4]

[1]Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Northfields Avenue, Wollongong NSW 2522, Australia (e-mail: qhl576@uowmail.edu.au, hduong@uow.edu.au, wsusilo@uow.edu.au)
[2]CSIRO Data61, Australia
[3]Faculty of Information Technology, Ho Chi Minh City University of Technology (HUTECH), 475A Dien Bien Phu, Ho Chi Minh City, Viet Nam
[3]Information Security Laboratory, KDDI Research, Inc., 2-1-15 Ohara, Fujimino-shi, Saitama, 356-8502, Japan (e-mail: ka-fukushima@kddi-research.jp, kiyomoto@kddi-research.jp)

Corresponding authors: Bay Vo (e-mail: vd.bay@hutech.edu.vn).

This work is partially supported by the ARC Linkage Project LP190100984. Huy Quoc Le has been sponsored by a CSIRO Data61 PhD Scholarship.

ABSTRACT Linkable ring signatures is a useful cryptographic tool for constructing applications such as ones relative to electronic voting (e-voting), digital cashes (e-cashes) as well as cloud computing. Equipped with linkable ring signatures, e-voting, e-cash systems can simultaneously enjoy the privacy and the unreusability properties thanks to the anonymity and the linkability of linkable ring signatures. Likewise, cloud servers can enjoy a privacy-preserving ability, a flexible access control and an efficient security management with linkable ring signatures. Moreover, linkable ring signatures built in the identity-based setting would help to remove the expense of using the conventional public key infrastructure and also could be applied to the user management. This primitive hence would be suitable for huge-scale applications. In this paper, we present the *first* identity-based linkable ring signatures (IdLRS) in both integer lattice and ideal lattice setting. The proposed IdLRS is proved secure in the random oracle model and based on the hardness of the short integer solution and ring short integer solution assumption. We also implement the proposed idLRS as a proof of concept and then do some experiments to evaluate the running times and the sizes.

INDEX TERMS Identity-based linkable ring signatures, e-voting, e-cash, cloud computing, lattices

I. INTRODUCTION

Two Main Issues in E-cash, E-voting. Digital cash and electronic voting are interesting applications of modern cryptography. Nowadays, while trading with digital cashes has become an undeniable and unstoppable trend, electronic voting has also been considered as a replacement of paper-based voting in many countries [20]. For the real-life usages, there are two common requirements for both e-voting systems and e-cash systems that they have to strongly offer the *Privacy* and *Unreusability* properties. On the one hand, the privacy property ensures that all voters' (resp., customers') identities to be anonymous. Seemingly, the blockchain mechanism being embedded in many cryptocurrencies, e.g., Bitcoin [35], Ethereum [11], Monero [36] etc., promises to provide the privacy ability. Unfortunately, in around 2012-2013, there were some reports on the Bitcoin's weak anonymity [6] [38], [42]. On the other hand, the unreusability property guarantees that no voters (resp., no coins) can vote (resp. can be spent) twice or more times. Particularly in e-cash, the unreusability property can be interpreted as being secure against the double-spending attacks. In fact, there was a report from the Bitcoin community relative to a double-spending attack against BetCoin Dice [16].

One Stone and Two Birds. Ring signatures (RS), first introduced by Rivest et al. [41], allow a member of a group of multiple signers to sign a message on the behalf of the group without revealing his identity. For more details, in ring signatures, there is no any group manager, there is no way to determine the identities of signers from some individual signature. Moreover, any set of signers in the group can be formed as a signing ring without any extra setup. Unfortunately, the strong anonymity of ring signatures may help an adversial signer to produce two (or more)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087808, IEEE Access

different signatures on the same message without being noticed. As a result, ring signature-based systems potentially lack the unreusability property. Aiming to avoid the problem, a relaxed variant of RS, called linkable ring signature (LRS), was then proposed by Liu et al. [26] in 2004, offering the *linkability* property. The linkability property allows one to link two signatures if these signatures are signed by the same signer[1], while still keeping the signer anonymous. As a consequence, LRS becomes a strong tool that can supports both the privacy-preserving ability and the unreusability as required by e-voting and e-cash systems. Actually, LRS has been embedded into many applications such as e-voting in [46], ad-hoc authentication in [26] and e-cash in [36], [37], [2], [1], [45], [50].

Application of LRS in Cloud Computing. Clouds are powerful resources that allow data owners to remotely store their data, to outsource heavy computations as well as to enjoy built-in services. On cloud service providers' (CSP) side, together with the requirement of protecting the privacy of users, a flexible access control and the efficient risk management are also challenging desires. As discussed in Liu et al. [25], linkable ring signatures are very suitable for cloud computing as they provides an anonymously dynamic access control mechanism and enhanced security control. For instance, using the linkability of LRS, CSPs are able to count the number of times a user has accessed, which helps to detect abnormal insider activities in the cloud system.

Identity-based Linkable Ring Signature and Huge-Scale Applications. The notion of identity-based (ID-based) cryptography, introduced by Shamir [43], aims to remove the dependency on the public key infrastructure (PKI) hence to simplify the certificate management in the traditional public key cryptography. In the ID-based cryptography, public key of each user is his identity (e.g., user name, email address, national identification number, domain name, physical IP address). This identity will be used to generate the private key for each user via a Private Key Generator using a master secret key. Instead of storing a list of certificates generated by PKI, ID-based cryptosystems just need to store the system parameters. In addition, the ID-based cryptography is also an effective method in managing user credentials. The ID-based cryptography also offer the delegation of decryption keys which is very useful in the case of applications having an enormous number of users [17].

Therefore, Identity-based linkable ring signatures (IdLRSs), LRSs that are built in the identity-based setting, would combine the advantages of both LRS and ID-based cryptography. As such, IdLRSs are very useful in huge-scale applications. For example, IdLRSs could be used for national elections having a huge number of voters, in which each voter can use his national identification number as identity. To reduce the storage and computing overheads, a national election system can delegate to its legitimate local election systems, which really manage the election with a much smaller number of local electors. One more example is that IdLRSs will also be appropriate for worldwide cloud systems having a vast number of users all over the world, in which each user is issued/registered with an identity. A worldwide cloud system can delegate its function to the country-wise branch cloud servers, to which users in each country really belong.

The security requirements for IdLRS are the anonymity, the unforgeability, the linkability and the nonslanderability. Informally speaking, the anonymity guarantees that the real signer is anonymous. The unforgeability prevents IdLRS from producing valid signatures by those who do not belong to the ring of signers. The nonslanderability requires that an adversary itself cannot produce a valid signature that is linked to a signature generated by an honest user. Following some existing works (e.g., [47], [5]) we consider the linkability of IdLRS in the event-oriented manner. In this manner, one can tell if two signatures are linked if and only if they are signed on the same event, even though they may be signed on behalf of different rings of signers. Event-oriented linkable ring signatures are comparatively more flexible in application and can helps to avoid some shortcomings group-oriented linkability [26]. A detailed discussion on group-oriented linkability and event-oriented linkability can be found in [47, Section 1].

Related Works. Linkable ring signature (LRS) was first proposed by Liu et al. [27] in 2005. Since then, there have been many follow-up works on this research line. e.g., [25], [2], [7], [10], [51], [48], [1], [28], [24], [49]. The first IdLRS was proposed in 2006 by Chow et al. [12]. From q-Strong Diffie-Hellman (q-SDH) and q-Decisional Strong Diffie-Hellman (q-DSDH) assumptions, the authors of [12] constructed an IdLRS instantiation which is secure in the random oracle model (ROM). In [12], those signatures, which were produced by the same signer in the same event, will be linked. In 2006, Au et al. [3] proposed a constant-size ID-based construction. However, later in 2009, Jeong et al. [21] made an analysis showing that the scheme [3] is insecure. In 2013, Au et al. [4] proposed a new ID-based event-oriented linkable ring signature scheme and prove the security of our scheme in the random oracle model, using the Discrete Logarithm (DL), the Decisional Diffie–Hellman (DDH) and q-Strong Diffie–Hellman (q-SDH) assumptions. Recently, in 2019, Deng et al. [13] have presented a new identity-based linkable ring signature scheme that is secure in ROM. The security of [13] is based on the hardness of the computational Diffie-Hellman (CDH) problem and the decisional bilinear Diffie-Hellman (DBDH) problem.

In Table 1, we demonstrate a summary of some existing identity-based linkable ring signatures in the literature. We

[1]Actually, many works such as [5], [12], [47] consider the linkability in an event-oriented manner, in the sense that two signatures is called linkable if they are signed on the same event and by the same signer. We will follow them in this work.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087808, IEEE Access

stress that, all works [12], [4] and [13] are provably secure based on classical number theory mathematical assumptions. Consequently, by Shor [44], they would be insecure against large-scale quantum computers. To the best of our knowledge, there have been no post-quantum secure IdLRS in the literature. Therefore, a post-quantum (e.g., lattice-based) IdLRS should be ideal and suitable for long-term applications.

Scheme	Assumption	Security Model	Post-quantum
Chow [12]	q-SDH, q-DSDH	ROM	×
Au [4]	DL, DDH, q-SDH	ROM	×
Deng [13]	CDH, DBDH	ROM	×
Ours	SIS	ROM	✓

TABLE 1: A summary of (ID-based) Linkable Ring Signatures in the literature.

Contribution and Overview. In this paper, we contribute to solve a long-standing open problem of lattice-based identity-based linkable ring signature by presenting the *first* quantum-secure IdLRS based on integer lattices (and ideal lattices). The proposed IdLRS enjoys the anonymity, the unforgeability, the linkability and the nonslanderability in the random oracle model.

In doing this, we combine the idea of ID-based ring signature [52] and the idea of linkable ring signature [2] to construct our IdLRS. A technically essential tool is the lattice trapdoor mechanism [33]. In our IdLRS construction, the public key and the master secret key are a matrix $\mathbf{A}$ and its $\mathbf{G}$-trapdoor $\mathbf{R}$ generated using a trapdoor generation. In order to extract the private key $\mathbf{S}_i$ for each user having identity id_i, using a secure hash function H_1, we let $\mathbf{A}_i = [\mathbf{A}|H_1(id_i)]$, then use the trapdoor delegation mechanism and finally sample $\mathbf{S}_i$ such that $\mathbf{A}_i\mathbf{S}_i = q\mathbf{I}_n \bmod 2q$ via a discrete Gaussian distribution. In the signing algorithm, for each signer having identity id_j other than the real one, a vector, say $\mathbf{z}_j$, will be chosen uniformly, whilst for the real signer of identity id_s, the rejection sampling is called to output $\mathbf{z}_s$ in such a way that $\mathbf{z}_s$ is independent of the private key $\mathbf{S}_s$ as well as looks like a uniform. Note that, we also use a secure hash function H_2 in the signing algorithm and use it in the same way as in [2]. Our IdLRS ensures that two messages are linked if they are produced by the same real signer in the same event. To this ends, we use a secure hash function H_3 to transform an event identity *event* into a matrix $\mathbf{K}$ and compute $\mathbf{E} = \mathbf{K}\mathbf{S}_s$, where $\mathbf{S}_s$ is the private key of the real signer in the ring.

We implement and do some experiments to give a proof of concept as well as to evaluate the practicability of the proposed IdLRS. The experimental results show that in the lattice-based IdLRS, the extraction algorithm is the most time consuming one. This is due to the inefficiency of the implemented Gaussian sampling algorithm over lattices (we implement this using the one in [19]). Note that, Gaussian sampling algorithms over lattices is still a bottleneck point in the lattice-based cryptography. However, we believe that, implementing with an appropriately chosen Gaussian sampling algorithm, the speed of the extraction algorithm will be accelerated.

We also remark that, we can improve the complexity and reduce the sizes of the proposed IdLRS by basing it on ideal lattices. We then also adapt the proposed IdLRS over integer lattices to get a version of IdLRS over ideal lattices (called rIdLRS) as presented in Section VI. Ideal lattices are ones with some additional algebraic structure. Specifically, ideal lattices corresponds to ideals in quotient rings of the form $\mathbb{Z}[x]/\langle f \rangle$ for some irreducible polynomial f. Thanks to the algebraic structure, ideal lattice-based cryptosystems enjoy high efficiency compared to integer lattice-based ones. We summarize the theoretical estimation of key sizes and signature size of our integer lattice-based IdLRS in Table 2. There, "$a \cdot S$" means "a elements in the set S". For setting parameters, the experimental results of running time and sizes, we refer the readers to Section V.

Paper Organization. In Section II, some background will be presented. We present the proposed construction in Section III. The security of the proposed IdLRS will be analysed in Section IV. Setting parameters, implementation and experimental results will be presented in Section V. An IdLRS construction based on the hardness of the ring SIS problem (rIdLRS) is also given in Section VI. We conclude this work in Section VII.

II. PRELIMINARIES

Norms. For any $\mathbf{R} = [\mathbf{r}_1|\cdots|\mathbf{r}_k] \in \mathbb{R}^{m\times k}$, denote $\widetilde{\mathbf{R}}$ to be the Gram-Schmidt orthogonalization (GSO) of $\mathbf{R}$. We will involve with the norms: (i) Euclidean norm: $\|\mathbf{R}\| := \max_i \|\mathbf{r}_i\|$, where $\|\mathbf{r}_i\|$ is the ordinary Euclidean norm; (ii) Gram-Schmidt (GS) norm: $\|\widetilde{\mathbf{R}}\|$; (iii) $\|\mathbf{x}\|_1 := \sum_{i=1}^n |x_i|$; and (iv) the sup norm: $s_1(\mathbf{R}) = \|\mathbf{R}\|_{\sup} = \sup_{\mathbf{x}} \frac{\|\mathbf{Rx}\|}{\|\mathbf{x}\|}$. Note that $s_1(\mathbf{R}) \geq \|\mathbf{R}\|$.

A. IDENTITY-BASED LINKABLE RING SIGNATURES

1) Syntax.

An identity-based Linkable Ring Signature (IdLRS) scheme is a tuple of efficient algorithms (IdLRS.Setup, IdLRS.Extract, IdLRS.Sign, IdLRS.Verify, IdLRS.Link) performing as follows:

- $(pp, msk) \leftarrow$ IdLRS.Setup(1^n): A probabilistic polynomial time (PPT) algorithm that takes as input a security parameter n to output public system parameters pp and a master secret key msk.
- $sk_{id} \leftarrow$ IdLRS.Extract(pp, id, msk): A PPT algorithm that takes as input public parameters pp, a user identity $id \in \{0,1\}^*$ and a master secret key msk, to generate a private key sk_{id} with respect to the identity id.
- $Sig \leftarrow$ IdLRS.Sign$(pp, event, \mu, id, \mathcal{R}, sk_{id})$: A PPT algorithm that on input public parameters pp, an event $event$, a message μ, a ring of signers $\mathcal{R}$, an identity id

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087808, IEEE Access

	Form	Size
Public key	$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$	$nm \cdot \mathbb{Z}_q$
Master secret key	$\mathbf{R} \sim D_{\sigma_1}^{\overline{m}\times nk}$	$\overline{m}nk \cdot D_{\sigma_1}$
Private key	$\mathbf{S}_i \sim D_{\sigma_3}^{(m+nk)\times n}$	$(m+nk)n \cdot D_{\sigma_3}$
Signature	$(\{\mathbf{z}_j\}_{j\in[\ell]}, \mathbf{c}_1, \mathcal{R}, \mathbf{E})$	$\ell(m+nk+n)\cdot D_\sigma + 1\cdot S_w^n + n^2 \cdot \mathbb{Z}_q$

TABLE 2: Theoretical estimation of key sizes and signature sizes for our integer lattice-based IdLRS version.

of the real signer in $\mathcal{R}$ and the corresponding private key sk_{id}, outputs a ring signature Sig.

- $1/0 \leftarrow$ IdLRS.Verify$(pp, event, \mu, Sig, \mathcal{R})$: A deterministic polynomial-time (DPT) that on input public parameters pp, an event $event$, a message μ, a ring of signers $\mathcal{R}$ and a ring signature Sig, returns 1 if the signature is valid, returns 0 otherwise.
- link/unlink $\leftarrow$ IdLRS.Link$((\mu_1, Sig_1), (\mu_2, Sig_2))$: A DPT algorithm that on input two valid message-signature pairs (μ_1, Sig_1), (μ_2, Sig_2), returns link if they are generated on the same event by the same signer, or returns unlink otherwise.

2) Correctness requirements.

- *Signing correctness:* Over the randomness of $(pp, msk) \leftarrow$ IdLRS.Setup(1^n), $sk_{id} \leftarrow$ IdLRS.Extract(pp, id, msk), and $Sig \leftarrow$ IdLRS.Sign$(pp, event, \mu, id, \mathcal{R}, sk_{id})$ then $\Pr[$IdLRS.Verify$(pp, event, \mu, Sig, \mathcal{R}) = 1] = 1 -$ negl(n).
- *Linking correctness:* Over the randomness of $(pp, msk) \leftarrow$ IdLRS.Setup(1^n), $sk_{id} \leftarrow$ IdLRS.Extract(pp, id, msk), $Sig_1 \leftarrow$ IdLRS.Sign$(pp, event, \mu_1, id, \mathcal{R}_1, sk_{id})$ and $Sig_2 \leftarrow$ IdLRS.Sign$(pp, event, \mu_2, id, \mathcal{R}_2, sk_{id})$, then $\Pr[$IdLRS.Link$((\mu_1, Sig_1), (\mu_2, Sig_2)) =$ link$] = 1 -$ negl(n).

3) Security models.

In order to state the security models for an IdLRS scheme, we summarise two kinds of queries that an adversary $\mathcal{A}$ can make in the corresponding games and the way the challenger responses to those queries.

- **Extract query** EQ(id_i): Once the adversary $\mathcal{A}$ makes an extract query on an identity id_i, the challenger $\mathcal{C}$ runs IdLRS.Extract(pp, id_i, msk) and hands $\mathcal{A}$ a private key sk_{id_i}.
- **Sign query** SQ$(\mu, event, \mathcal{R}, id_s)$: Once the adversary $\mathcal{A}$ makes a sign query on a tuple of $(\mu, event, \mathcal{R}, id_s \in \mathcal{R})$, the challenger $\mathcal{C}$ first computes sk_{id_s} by IdLRS.Extract(pp, id_s, msk) and then sends $Sig \leftarrow$ IdLRS.Sign$(pp, event, \mu, id_s, \mathcal{R}, sk_{id_s})$ to $\mathcal{A}$.

Definition 1 (Anonymity)**.** *Anonymity of an IdLRS ensures that from a valid ring signature, it is impossible (for any adversary) to decide who the real signer is. Formally, an IdLRS scheme is called to be anonymous if for any polynomial-time adversary $\mathcal{A}$ playing in GAME I below, the probability that $\mathcal{A}$ wins is negligible.*

GAME I (Anonymity Game):

- ***Setup.*** *Given n, the challenger $\mathcal{C}$ calls the algorithm* IdLRS.Setup(1^n) *to get public parameters pp and a master secret key msk. Then $\mathcal{C}$ sends pp to the adversary $\mathcal{A}$.*
- ***Query 1.*** *$\mathcal{A}$ adaptively makes a polynomially bounded number of extract queries EQ(id_i) and sign queries SQ($\mu, event, \mathcal{R}, id_s$), and the challenger responses in such a way mentioned above.*
- ***Challenge.*** *$\mathcal{A}$ submits a message μ, an event $event$, a ring $\mathcal{R}$, and two identities $id_{s_0}, id_{s_1} \in \mathcal{R}$ such that EQ(id_{s_0}), EQ(id_{s_1}), SQ($\cdot, event, \cdot, id_{s_0}$) and SQ($\cdot, event, \cdot, id_{s_1}$) have not been queried before. Now, $\mathcal{C}$ chooses randomly $b \xleftarrow{\$} \{0,1\}$, generates $sk_{id_{s_b}}$ by IdLRS.Extract(pp, id_{s_b}, msk) and returns $Sig \leftarrow$* IdLRS.Sign$(pp, event, \mu, id_{s_b}, \mathcal{R}, sk_{id_{s_b}})$.
- ***Query 2.*** *Same as **Query 1**, except that $\mathcal{A}$ is not allowed to make queries EQ(id_{s_0}), EQ(id_{s_1}), SQ($\cdot, event, \cdot, id_{s_0}$) and SQ($\cdot, event, \cdot, id_{s_1}$).*
- ***Guess.*** *The adversary $\mathcal{A}$ outputs a guess b' for b. $\mathcal{A}$ wins if $b' = b$.*

Definition 2 (Unforgeability)**.** *Unforgeability of an IdLRS guarantees that any one, who does not have any private key of signers in some ring, cannot produce a valid ring signature on that ring. Formally, an IdLRS scheme is unforgeable under adaptive chosen-identity and chosen-massage attacks if, for any polynomial-time adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins GAME II below is negligible.*

GAME II (Unforgeability Game):

- ***Setup.*** *Same as GAME I.*
- ***Query.*** *Same as **Query 1** of GAME I.*
- ***Forge.*** *Eventually, $\mathcal{A}$ outputs a ring signature Sig^* on a message μ^*, an event $event^*$ and a ring $\mathcal{R}^*$. It wins the game if:*
  1) *$(\mu^*, event^*, \mathcal{R}^*, Sig^*)$ is valid, that is, IdLRS.Verify($pp, event^*, \mu^*, Sig^*, \mathcal{R}^*$)=1.*
  2) *Sign queries SQ($\mu^*, event^*, \mathcal{R}^*, id$), $\forall id \in \mathcal{R}^*$ and extract queries EQ(id), $\forall id \in \mathcal{R}^*$ have never been made in the **Query** phase.*

Definition 3 (Linkability)**.** *Linkability of an IDLRS requires that two different ring signatures produced on the same event and by the same real signer who belongs to two (unnecessarily same) rings must be linkable. Formally, an IdLRS scheme is linkable for the same event if for any polynomial-time adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins GAME III below*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087808, IEEE Access

is negligible.

GAME III (Linkability Game):

- ***Setup.*** *Same as GAME I.*
- ***Query.*** *Same as **Query 1** of GAME I.*
- ***Unlink.*** *Finally, $\mathcal{A}$ outputs two tuples $(\mu_1, event, \mathcal{R}_1, Sig_1)$ and $(\mu_2, event, \mathcal{R}_2, Sig_2)$ on the same event. The adversary wins if all the following conditions hold:*
  1) $(\mu_1, event, \mathcal{R}_1, Sig_1)$ *and* $(\mu_2, event, \mathcal{R}_2, Sig_2)$ *are valid.*
  2) $(\mu_1, event, \mathcal{R}_1, Sig_1)$ *and* $(\mu_2, event, \mathcal{R}_2, Sig_2)$ *are not obtained through the **Query** phase.*
  3) $\mathcal{A}$ *is given at most one private key* sk_{id}, *with* $id \in \mathcal{R}_1 \cup \mathcal{R}_2$.
  4) $\mathsf{IdLRS.Link}(Sig_1, Sig_2) = \mathsf{unlinked}$.

Definition 4 (Nonslanderability)**.** *Nonslanderability of an IDLRS ensures that any adversary without having the private key of the real signer in the ring cannot produce any new signatures that are linkable to the previous ones. Formally, an IdLRS scheme is nonslanderable for the same event if for any polynomial-time adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins GAME IV is negligible.*

GAME IV (Nonslanderability Game):

- ***Setup.*** *Same as GAME I.*
- ***Query 1.*** *Same as **Query 1** of GAME I.*
- ***Challenge.*** *The adversary $\mathcal{A}$ submits a tuple of $(\mu, event, \mathcal{R}, id^* \in \mathcal{R})$, such that EQ($id^*$) has not been queried before. The challenger $\mathcal{C}$ generates sk_{id^*} by IdLRS.Extract(pp, id^*, msk) and returns $Sig \leftarrow$ $\mathsf{IdLRS.Sign}(pp, event, \mu, id^*, \mathcal{R}, sk_{id^*})$.*
- ***Query 2.*** *Same as **Query 1**, except that $\mathcal{A}$ is not allowed to make queries EQ(id^*), and SQ($\cdot, event, \cdot, id^*$).*
- ***Slander.*** *The adversary $\mathcal{A}$ outputs a new signature Sig' on the same message μ and the same event event. The adversary wins if the following conditions hold:*
  1) $(\mu, event, \mathcal{R}', Sig')$ *is valid.*
  2) $(\mu, event, \mathcal{R}', Sig')$ *was not obtained through **Query 1** and **Query 2**.*
  3) $\mathsf{IdLRS.Link}(Sig, Sig') = \mathsf{linked}$.

Constructing IdLRS in the lattice setting is a long-standing problem. In this paper, we will give the first lattice-based IdlRS constructions. In the next section, we will review some background of lattices.

B. BACKGROUND OF LATTICES

Lattices. A lattice Λ in $\mathbb{Z}^m$ is a set of integral combinations of given some linearly independent vectors, say $\{\mathbf{b}_1, \cdots, \mathbf{b}_n\} \subset \mathbb{Z}^m$, which is formally defined as

$$\Lambda := \left\{ \sum_{i=1}^{n} \mathbf{b}_i x_i | x_i \in \mathbb{Z}\ \forall i = 1, \cdots, n \right\} \subseteq \mathbb{Z}^m.$$

Given a matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$ and a vector $\mathbf{u} \in \mathbb{Z}_q^n$, one can prove that the following sets are essentially lattices:

$$\Lambda_q(\mathbf{A}) := \left\{ \mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \exists \mathbf{s} \in \mathbb{Z}_q^n \text{ where } \mathbf{A}^T\mathbf{s} = \mathbf{e} \bmod q \right\},$$
$$\Lambda_q^{\perp}(\mathbf{A}) := \{ \mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \mathbf{A}\mathbf{e} = \mathbf{0} \bmod q \},$$
$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) := \{ \mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \mathbf{A}\mathbf{e} = \mathbf{u} \bmod q \}.$$

Hardness Assumption. The short integer solution (SIS) problem is an average hard problem in lattices on which we rely our proposed scheme's security. The problem is stated as follows:

Definition 5 (SIS Problem)**.** *Given positive integers q, m, a random matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and $\beta \in \mathbb{R}^+$, the $\mathsf{SIS}_{n,m,q,\beta}$ problem requires to seek a non-zero short vector $\mathbf{e} \in \mathbb{Z}^m$ satisfying $\|\mathbf{e}\| \leq \beta$ and $\mathbf{A}\mathbf{e} = \mathbf{0} \pmod q$.*

The following lemmas present the hardness of SIS problem as well as the condition for which the $\mathsf{SIS}_{n,m,q,\beta}$ problem has a solution.

Lemma 1 ([19, Proposition 5.7])**.** *For any poly-bounded m, and $\beta = \mathsf{poly}(n)$, and for any prime $q \geq \beta \cdot \omega(\sqrt{n \log n})$, average-case $\mathsf{SIS}_{n,m,q,\beta}$ and $\mathsf{ISIS}_{n,m,q,\beta}$ is as hard as SIVP_γ (among others) in the worst-case to within certain $\gamma = \widetilde{O}(\beta\sqrt{n})$ factor.*

Lemma 2 ([34, Lemma 5.2])**.** *For any q, $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and $\beta \geq \sqrt{m} q^{n/m}$, the $\mathsf{SIS}_{n,m,q,\beta}$ admits a solution.*

Smoothing Parameters. Smoothing parameters is proposed to measure the quality of a lattice by Micianco and Regev [34].

Definition 6 (Smoothing Parameters, [34])**.** *For any n-dimensional lattice $\mathcal{L}$ and positive real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ is the smallest real number $s > 0$ such that $\rho_{1/s}(\mathcal{L}^* \setminus \{\mathbf{0}\}) \leq \epsilon$.*

Note that, for any $\epsilon \in (0,1)$, $\eta_\epsilon(\mathbb{Z}) = \sqrt{\frac{\ln(2(1+1/\epsilon))}{\pi}}$.

Discrete Gaussians. Define: $\rho_{\mathbf{c},\sigma}(\mathbf{x}) = \exp\left(-\pi\frac{\|\mathbf{x}-\mathbf{c}\|^2}{\sigma^2}\right)$, $\rho_{\mathbf{c},\sigma}(\Lambda) = \sum_{\mathbf{x}\in\Lambda} \rho_{\mathbf{c},\sigma}(\mathbf{x})$, where $\Lambda \subseteq \mathbb{Z}^m$ is a lattice, $\mathbf{c} \in \mathbb{R}^m$ and a positive parameter $\sigma > 0$. The discrete Gaussian distribution over Λ with center $\mathbf{c}$ and parameter σ is defined by the function $D_{\Lambda,\mathbf{c},\sigma}(\mathbf{y}) = \frac{\rho_{\mathbf{c},\sigma}(\mathbf{y})}{\rho_{\mathbf{c},\sigma}(\Lambda)}$, where $\mathbf{y} \in \Lambda$. If $\mathbf{c} = \mathbf{0}$, we drop it out for convenience, i.e., we just write ρ_σ and $D_{\Lambda.\sigma}$ standing for $\rho_{\sigma,\mathbf{0}}$ and $D_{\Lambda,\mathbf{0},\sigma}$ respectively. If $\Lambda = \mathbb{Z}^m$, we just write $D^m_{\mathbf{c},\sigma}$ instead of $D_{\mathbb{Z}^m,\mathbf{c},\sigma}$. For $\sigma = 1$, we will use ρ as a replacement of ρ_1.

Lemma 3 ([34])**.** *Let n, q be any positive integers and $m \geq 2n \log q$. Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a basis $\mathbf{B}$ of $\Lambda_q^{\perp}(\mathbf{A})$ and a vector $\mathbf{u} \in \mathbb{Z}_q^n$, if the Gaussian parameter $\sigma \geq \|\widetilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log n})$, then for any $\mathbf{x} \xleftarrow{\$} D_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma}$, we have $\|\mathbf{x}\| \leq \sigma\sqrt{m}$ with overwhelming probability.*

Lemma 4 ([29, Lemma 4.4-4.5])**.** *For any positive $\eta, \sigma \in \mathbb{R}$ any vector $\mathbf{c} \in \mathbb{Z}^m$, we have*

1) $\Pr[|z| > \eta\sigma; z \xleftarrow{\$} D^1_\sigma] \leq 2e^{\frac{-\eta^2}{2}}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087808, IEEE Access

2) *For* $\eta > 1$, $\Pr[\|\mathbf{z}\| > \eta\sigma\sqrt{m}; \mathbf{z} \xleftarrow{\$} D_\sigma^m] < \eta^m e^{\frac{m}{2}(1-\eta^2)}$.
3) *For* $\mathbf{z} \in \mathbb{Z}^m$, *if* $\sigma \geq 3/\sqrt{2\pi}$, *then* $D_\sigma^m(\mathbf{z}) \leq 2^{-m}$.
4) $\Pr[D_\sigma^m(\mathbf{z})/D_{\mathbf{c},\sigma}^m(\mathbf{z}) = O(1); \mathbf{z} \xleftarrow{\$} D_\sigma^m] = 1 - 2^{-\omega(\log m)}$ *for* $\sigma = \omega(\|\mathbf{c}\|\sqrt{\log m})$. *Specifically, for any* $\mathbf{c} \in \mathbb{Z}^m$, *if* $\sigma = \alpha \cdot \|\mathbf{c}\|$, *where* $\alpha > 0$, *we have*

$$\Pr\left[\frac{\mathcal{D}_\sigma^m(\mathbf{x})}{\mathcal{D}_{\mathbf{c},\sigma}^m(\mathbf{x})} \leq e^{12/\alpha+1/(2\alpha^2)} : \mathbf{x} \leftarrow \mathcal{D}_\sigma^m\right] \geq 1-2^{-100}.$$

Remark 1. *In Item 2 of Lemma 4, one usually chooses* $\eta \in [1.1, 1.3]$. *(See [23, Remark 2] for a detailed discussion.)*

Remark 2. *In Item 4 of Lemma 4, if* $\alpha = 12$, *i.e.,* $\sigma = 12\|\mathbf{c}\|$ *then with probability at least* $1 - 2^{-100}$, *we have* $\mathcal{D}_\sigma^m(\mathbf{x})/\mathcal{D}_{\mathbf{c},\sigma}^m(\mathbf{x}) \leq e^{1+1/288}$.

C. G-TRAPDOORS AND RELATED ALGORITHMS

In this section, we will present the notion of $\mathbf{G}$-trapdoor and recall a special matrix, called *primitive matrix* [33], which will play an important role in our scheme.

Definition 7 (**G**-trapdoors, [33, Definittion 5.2])**.** *Let* n, q, m, k *be positive integers and* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{G} \in \mathbb{Z}_q^{n\times nk}$ *be matrices with* $m \geq nk$. *Let* $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ *be some invertible matrix. The* $\mathbf{G}$*-trapdoor for* $\mathbf{A}$ *with tag* $\mathbf{H}$ *is a matrix* $\mathbf{R} \in \mathbb{Z}^{(m-nk)\times nk}$ *such that* $\mathbf{A}\left[\begin{smallmatrix}\mathbf{R}\\ \mathbf{I}_{nk}\end{smallmatrix}\right] = \mathbf{HG} \pmod{q}$.

The quality of the $\mathbf{G}$-trapdoor $\mathbf{R}$ is measured by its largest singular value $s_1(\mathbf{R})$, which is essentially small if every element of $\mathbf{R}$ is sampled from D_σ. Formally we have the following lemma.

Lemma 5 ([33, Lemma 2.9])**.** *Let* $D_\sigma^{n\times m}$ *be a discrete Gaussian distribution with parameter* σ *and* $\mathbf{R} \leftarrow D_\sigma^{n\times m}$. *Then with overwhelming probability* $s_1(\mathbf{R}) \leq \sigma \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{n}+\sqrt{m})$.

In particular, we only focus on the primitive matrix $\mathbf{G}$ defined as $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^t \in \mathbb{Z}_q^{n\times nk}$, where $k = \lceil \log_2 q \rceil$, $\mathbf{g}^t = (1, 2, 4, ..., 2^{k-1}) \in \mathbf{Z}_q^k$, $\mathbf{I}_n \in \mathbb{Z}^{n\times n}$ is an identity matrix and $\otimes$ stands for the tensor product. Moreover, one can find a short special basis, say $\mathbf{B}_k \in \mathbb{Z}^{k\times k}$ for $\Lambda^\perp(\mathbf{g}^t)$, i.e., $\mathbf{g}^t.\mathbf{B}_k = \mathbf{0} \in \mathbb{Z}_q^k$. Accordingly, the short matrix $\mathbf{B} := \mathbf{I}_n \otimes \mathbf{B}_k \in \mathbb{Z}^{nk\times nk}$ is the basis of $\Lambda^\perp(\mathbf{G})$.

Now we recall several useful algorithms related to the primitive matrix $\mathbf{G}$ and $\mathbf{G}$-trapdoors. In the following, let $q \geq 2, \overline{m} \geq 1, k = \lceil \log_2 q \rceil$, and $m = O(n \log q)$.

- $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{GenTrap}(\overline{\mathbf{A}}, \mathbf{H}, \sigma)$ [33, Algorithm 1]: Given a uniformly random matrix $\overline{\mathbf{A}} \in \mathbb{Z}_q^{n\times\overline{m}}$ and an invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$, the polynomial time algorithm $\mathsf{GenTrap}(\overline{\mathbf{A}}, \mathbf{H})$ will output a random matrix $\mathbf{A} = \left[\overline{\mathbf{A}}|\mathbf{HG} - \overline{\mathbf{A}}\mathbf{R}\right]$ and a $\mathbf{G}$-trapdoor $\mathbf{R} \sim D_\sigma^{\overline{m}\times nk}$ with tag $\mathbf{H}$, where $\sigma \geq \eta_\epsilon(\mathbb{Z})$ for any $\epsilon \in (0, 1)$ (we should choose $\sigma \geq \sqrt{\frac{\ln(2(1+1/\epsilon))}{\pi}})$. Note that, there exists $\epsilon = \epsilon(n)$ negligible for which $\sigma \geq \omega(\sqrt{\log n})$. Also, by Lemma 5, $s_1(\mathbf{R}) \leq \sigma \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{\overline{m}} + \sqrt{nk})$.
- $\mathbf{e} \leftarrow \mathsf{SampleD}(\mathbf{A}, \mathbf{R}, \mathbf{H}, \mathbf{u}, \sigma)$ [33, Algorithm 3]: Given a $\mathbf{G}$-trapdoor $\mathbf{R} \in \mathbb{Z}^{\overline{m}\times nk}$ for $\mathbf{A} \in \mathbb{Z}_q^{n\times(\overline{m}+nk)}$, an invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$, a uniform vector $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$ and Gaussian parameter $\sigma \geq \sqrt{7(s_1(\mathbf{R})^2 + 1)} \cdot \omega(\sqrt{\log n})$ (see [33, Section 5.4]). The polynomial time algorithm $\mathsf{SampleD}(\mathbf{A}, \mathbf{R}, \mathbf{H}, \mathbf{u}, \sigma)$ will output a vector $\mathbf{e} \in \mathbb{Z}^{m+nk}$ sampled from a distribution that is statistically close to $D_{\Lambda^{\mathbf{u}}(\mathbf{A}),\sigma}$.
- $\mathbf{R}' \leftarrow \mathsf{DelTrap}(\mathbf{A}, \mathbf{A}_1, \mathbf{R}, \mathbf{H}, \sigma)$ [33, Algorithm 4]: Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ along with a $\mathbf{G}$-trapdoor $\mathbf{R} \in \mathbb{Z}^{(m-nk)\times nk}$, a new matrix $\mathbf{A}_1 \in \mathbb{Z}_q^{n\times nk}$, an invertible matrix $\mathbf{H} \in \mathbb{Z}^{n\times n}$ and a Gaussian parameter $\sigma \geq \eta_\epsilon(\Lambda_q^\perp(\mathbf{A}))$, (we should choose $\sigma \geq \sqrt{5}(s_1(\mathbf{R}) + 1) \cdot \omega(\sqrt{\log n})$ (see [33, Lemma 2.3, Lemma 5.3])), the polynomial time algorithm $\mathsf{DelTrap}(\mathbf{A}, \mathbf{A}_1, \mathbf{R}, \mathbf{H}, \sigma)$ will output a $\mathbf{G}$-trapdoor $\mathbf{R}' \in \mathbb{Z}^{m\times nk}$ for matrix $[\mathbf{A}|\mathbf{A}_1]$ with tag $\mathbf{H}$.

In the rest of the paper, we will set $\mathbf{H} = \mathbf{I}_n$ and omit it for simplicity.

D. REJECTION SAMPLING

Lemma 6 (**Rejection Sampling**, [14])**.** *Let* m *be a positive integer and* V *be an arbitrary set. Let* $f : \mathbb{Z}^m \to \mathbb{R}$ *be probability distributions. If* $g_v : \mathbb{Z}^m \to \mathbb{R}$ *is a family of probability distributions indexed by* $v \in V$ *with the property that*

$$\exists M \in \mathbb{R}^+ \; s.t \; \forall v \in V, \Pr[M \cdot g_v(\mathbf{z}) \geq f(\mathbf{z}); z \xleftarrow{\$} f] \geq 1 - \epsilon$$

Then the distributions of the following two algorithms are statistically indistinguishable (within statistical distance $\Delta(\mathcal{A}, \mathcal{F}) = \frac{\epsilon}{M}$ *).*

1) $\mathcal{A}$*:* $v \leftarrow h, \mathbf{z} \leftarrow g_v$, *output*$(\mathbf{z}, v)$ *with probability* $f(\mathbf{z})/(M \cdot g_v(\mathbf{z}))$*;*
2) $\mathcal{F}$*:* $v \leftarrow h, \mathbf{z} \leftarrow f$, *output*$(\mathbf{z}, v)$ *with probability* $1/M$.

III. OUR IDENTITY-BASED LINKABLE RING SIGNATURE SCHEME

A. DESCRIPTION

In our scheme, each signer of a ring has an identity id. For simplicity, we denote a ring by a tuple of identity, e.g., $\mathcal{R} = (id_1, \cdots, id_\ell)$. From now on, we always consider $\mathbf{c}_i = \mathbf{c}_{i \bmod \ell}$. Our scheme consists of algorithms IdLRS.Setup, IdLRS.Extract, IdLRS.Sign, IdLRS.Verify and IdLRS.Link working as follows:

IdLRS.Setup(1^n): On input a security parameter n, do the following:

1) Choose integers $q \geq 2$, $w \geq 3$, $M \leq 3$ fixed and $k := \lceil \log(2q) \rceil$ and $\overline{m} \geq 1$, such that $m := \overline{m} + nk \geq O(n \log q)$.
2) Choose $\sigma_1, \sigma_2, \sigma_3, \sigma$ to be Gaussian parameters.
3) Choose three collision-resistant hash functions $H_1 : \{0, 1\}^* \to \mathbb{Z}_q^{n\times nk}$, $H_2 : \{0, 1\}^* \to S_w^n$, where $S_w^n := \{\mathbf{c} \in \{0, 1\}^n : \|\mathbf{c}\|_1 = w\}$, and $H_3 : \{0, 1\}^* \to \mathbb{Z}_q^{n\times(m+nk)}$. These hash functions will play as random oracles in our security proof later. This means that their outputs look like random.

4) Choose $\overline{\mathbf{A}} \xleftarrow{\$} \mathbb{Z}_q^{n\times nk}$, and run $\mathsf{GenTrap}(\overline{\mathbf{A}}, \mathbf{I}, \sigma_1)$ to get a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ along with a $\mathbf{G}$-trapdoor $\mathbf{R} \sim D_{\sigma_1}^{\overline{m}\times nk}$.
5) The public key is $pk := \mathbf{A}$ and the master secret key is $msk := \mathbf{R}$ and system public parameter pp consists of H_1, H_2, H_3 and the rest parameters.

$\mathsf{IdLRS.Extract}(id_i, msk)$: On input an identity $id_i \in \{0,1\}^*$ of a user in a ring and a master secret key $msk = \mathbf{R}$, do:

1) Compute $\mathbf{Q}_i = H_1(id_i)$ and set $\mathbf{A}_i := [\mathbf{A}|\mathbf{Q}_i] \in \mathbb{Z}_q^{n\times(m+nk)}$.
2) Sample $\mathbf{R}_i \leftarrow \mathsf{DelTrap}(\mathbf{A}, \mathbf{Q}_i, \mathbf{R}, \sigma_2)$, $\mathbf{R}_i \sim D_{\sigma_2}^{m\times nk}$.
3) For $t \in [n]$, sample $\mathbf{s}_{i,t} \in \mathbb{Z}^{(m+nk)\times n} \leftarrow \mathsf{SampleD}(\mathbf{A}_i, \mathbf{R}_i, \mathbf{I}_n, \mathbf{u}_t, \sigma_3)$ such that $\mathbf{A}_i\mathbf{s}_{i,t} = \mathbf{u}_t \bmod q$, where $\mathbf{u}_t$ is the t-th column of $q\mathbf{I}_n$. Let $\mathbf{S}_i \in \mathbb{Z}^{(m+nk)\times n} = [\mathbf{s}_{i,1}|\cdots|\mathbf{s}_{i,n}]$. Then, $\mathbf{A}_i\mathbf{S}_i = q\mathbf{I}_n \bmod 2q$.
4) Output the private key $sk_{id_i} := \mathbf{S}_i$.

$\mathsf{IdLRS.Sign}(\mu, event, \mathcal{R}, sk_s)$: On input a message μ, an event $event$, a ring of ℓ users $\mathcal{R} = (id_1, ..., id_\ell)$, an identity $id_s \in \mathcal{R}$ and a corresponding key $sk_s = \mathbf{S}_s$, do:

1) Let $\mathbf{K} := H_3(event)$ and $\mathbf{E} := \mathbf{K}\mathbf{S}_s \in \mathbb{Z}_q^{n\times n}$.
2) Let $\widehat{\mathbf{K}} \leftarrow [2\mathbf{K}|-2\mathbf{E}+q\mathbf{I}_n] \in \mathbb{Z}_{2q}^{n\times(m+nk+n)}$ and $\widehat{\mathbf{S}}_s \leftarrow \left[\begin{smallmatrix}\mathbf{S}_s\\ \mathbf{I}_n\end{smallmatrix}\right] \in \mathbb{Z}_{2q}^{(m+nk+n)\times n}$. Note that $\widehat{\mathbf{K}}\cdot\widehat{\mathbf{S}}_s = q\mathbf{I}_n \bmod 2q$.
3) For $i \in [\ell]$, let $\widehat{\mathbf{A}}_i \leftarrow [\mathbf{A}_i|\mathbf{0}] \in \mathbb{Z}_{2q}^{n\times(m+nk+n)}$. Then, $\widehat{\mathbf{A}}_s \cdot \widehat{\mathbf{S}}_s = q\mathbf{I}_n \bmod 2q$.
4) Choose a vector $\mathbf{y} \leftarrow D_\sigma^{m+nk+n}$.
5) Calculate $\mathbf{c}_{s+1} = H_2(\widehat{\mathbf{A}}_s\mathbf{y} \bmod 2q,\ \widehat{\mathbf{K}}\mathbf{y} \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E})$.
6) For each identity $id_j \in \mathcal{R}\setminus\{id_s\}$, choose a vector $\mathbf{z}_j \leftarrow D_\sigma^{m+nk+n}$.
7) For $i = s+1, \cdots, \ell-1, 0, 1, \cdots, s-1$, do:
   - Calculate $\mathbf{c}_{i+1} = H_2(\widehat{\mathbf{A}}_i\mathbf{z}_i + q\mathbf{c}_i \bmod 2q, \widehat{\mathbf{K}}\mathbf{z}_i + q\mathbf{c}_i \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E})$.
8) For $j = s$, choose $b \xleftarrow{\$} \{0,1\}$ and calculate $\mathbf{z}_s \leftarrow (-1)^b\widehat{\mathbf{S}}_s\mathbf{c}_s + \mathbf{y} \bmod 2q$ and output $\mathbf{z}_s$ with probability $\min\left\{\frac{D_\sigma^{m+nk+n}(\mathbf{z}_s)}{M\cdot D^{m+nk+n}_{(-1)^b\widehat{\mathbf{S}}_s\mathbf{c}_s,\sigma}(\mathbf{z}_s)}, 1\right\}$.
9) Output the ring signature $\sigma_\mathcal{R} = \sigma_\mathcal{R}(\mu, event) = (\{\mathbf{z}_j\}_{j\in[\ell]}, \mathbf{c}_1, \mathcal{R}, \mathbf{E})$.

$\mathsf{IdLRS.Verify}(\mu, event, \sigma_\mathcal{R})$: Take as input a message μ, an event $event$ and a signature $\sigma_\mathcal{R} = (\{\mathbf{z}_j\}_{j\in[\ell]}, \mathbf{c}_1, \mathcal{R}, \mathbf{E})$, do the following:

1) If for all $j \in [\ell]$, $\|\mathbf{z}_j\| \le \Delta := \eta\sigma\sqrt{m+nk+n}$ where $1.1 \le \eta \le 1.3$ go to Step 2; otherwise output 0.
2) Let $\mathbf{K} = H_3(event)$, and let $\widehat{\mathbf{K}} \leftarrow [\mathbf{K}| - \mathbf{E} + q\mathbf{I}_n] \in \mathbb{Z}_{2q}^{n\times(m+nk+n)}$.
3) For $i \in [\ell-1]$, do:
   - Let $\widehat{\mathbf{A}}_i \leftarrow [\mathbf{A}_i|\mathbf{0}] \in \mathbb{Z}_{2q}^{n\times(m+nk+n)}$,
   - Calculate $\mathbf{c}_{i+1} = H_2(\widehat{\mathbf{A}}_i\mathbf{z}_i + q\mathbf{c}_i \bmod 2q, \widehat{\mathbf{K}}\mathbf{z}_i + q\mathbf{c}_i \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E})$.
4) If $\mathbf{c}_1 = H_2(\widehat{\mathbf{A}}_\ell\mathbf{z}_\ell + q\mathbf{c}_\ell \bmod 2q, \widehat{\mathbf{K}}\mathbf{z}_\ell + q\mathbf{c}_\ell \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E})$ output 1, otherwise output 0.

$\mathsf{IdLRS.Link}(\sigma_{\mathcal{R}_1}, \sigma_{\mathcal{R}_2})$: Take as input two sing signatures $(\sigma_{\mathcal{R}_1} = (\{\mathbf{z}_{1,j}\}_{j\in[\ell]}, \mathbf{c}_{1,1}, \mathcal{R}_1, \mathbf{E}_1))$ and $(\sigma_{\mathcal{R}_2} = (\{\mathbf{z}_{2,j}\}_{j\in[\ell']}, \mathbf{c}_{1,2}, \mathcal{R}_2, \mathbf{E}_2))$, perform:

1) Output linked if both $\sigma_{\mathcal{R}_1}$ and $\sigma_{\mathcal{R}_2}$ are valid and $\mathbf{E}_1 = \mathbf{E}_2$. Otherwise output unlinked.

B. CORRECTNESS

1) Signing Correctness.

For the signing correctness, we need to show that $H_2(\widehat{\mathbf{A}}_\ell\mathbf{z}_\ell + q\mathbf{c}_\ell \bmod 2q, \widehat{\mathbf{K}}\mathbf{z}_\ell + q\mathbf{c}_\ell \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E}) = \mathbf{c}_1$ (in IdLRS.Verify) and $H_2(\widehat{\mathbf{A}}_i\mathbf{z}_i + q\mathbf{c}_i \bmod 2q, \widehat{\mathbf{K}}\mathbf{z}_i + q\mathbf{c}_i \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E}) = \mathbf{c}_{i+1}$ for $1 \le i \le \ell-1$ (in IdLRS.Sign). Suppose that id_s is the identity of the real signer in $\mathcal{R}$. Then we have two cases:

- If $i \ne s$, then $\mathbf{c}_{i+1} = H_2(\widehat{\mathbf{A}}_i\mathbf{z}_i + q\mathbf{c}_i, \widehat{\mathbf{K}}\mathbf{z}_i + q\mathbf{c}_i, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E})$ is the same for both IdLRS.Sign and IdLRS.Verify.
- For $i = s$, remind that in IdLRS.Sign we have $\mathbf{c}_{s+1} \leftarrow H_2(\widehat{\mathbf{A}}_s\mathbf{y} \bmod 2q, \widehat{\mathbf{K}}\mathbf{y} \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E})$, whilst in IdLRS.Verify, $\mathbf{c}_{s+1} \leftarrow H_2(\widehat{\mathbf{A}}_s\mathbf{z}_s + q\mathbf{c}_s \bmod 2q, \widehat{\mathbf{K}}\mathbf{z}_s + q\mathbf{c}_s \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E})$. We will prove that $\mathbf{c}_{s+1}$ (in IdLRS.Sign) $= \mathbf{c}_{s+1}$ (in IdLRS.Verify). Indeed, we can obtain $\widehat{\mathbf{A}}_s\mathbf{y} = \widehat{\mathbf{A}}_s\mathbf{z}_s + q\mathbf{c}_s$, which is equivalent to $\widehat{\mathbf{A}}_s(\mathbf{y} - \mathbf{z}_s) = q\mathbf{c}_s$, by replacing $\mathbf{z}_s$ with $(-1)^b\widehat{\mathbf{S}}_s\mathbf{c}_s + \mathbf{y}$ to get
$$-(-1)^b\widehat{\mathbf{A}}_s\widehat{\mathbf{S}}_s\mathbf{c}_s = q\mathbf{c}_s \bmod 2q, \text{ i.e.,}$$
$$-(-1)^b q\mathbf{c}_s = q\mathbf{c}_s \bmod 2q.$$
Clearly, this equation holds for all $b \in \{0,1\}$ thanks to $q\mathbf{c}_s = \pm q\mathbf{c}_s \bmod 2q$.
Similarly, we also have $\widehat{\mathbf{K}}\mathbf{y} = \widehat{\mathbf{K}}\mathbf{z}_s + q\mathbf{c}_s \bmod 2q$.

2) Linking Correctness.

We consider two valid ring signatures $\sigma_{\mathcal{R}_1} = (\{\mathbf{z}_{1,j}\}_{j\in[\ell]}, \mathbf{c}_{1,1}, \mathcal{R}_1, \mathbf{E}_1)$ and $\sigma_{\mathcal{R}_2} = (\{\mathbf{z}_{2,j}\}_{j\in[\ell']}, \mathbf{c}_{1,2}, \mathcal{R}_2, \mathbf{E}_2)$, in which an honest user of identity $id_s \in \mathcal{R}_1 \cap \mathcal{R}_2$ is the real signer, signing on two messages μ_1 and μ_2 and on the same event $event$. Then $\mathsf{IdLRS.Link}(\sigma_{\mathcal{R}_1}, \sigma_{\mathcal{R}_2})$ outputs linked with overwhelming probability. Indeed, the facts that $\Pr[\mathbf{E}_1 = \mathbf{E}_2] = \Pr[\mathbf{K}_1\mathbf{S}_s = \mathbf{K}_2\mathbf{S}_s]$ and that $\mathbf{K}_1 = \mathbf{K}_2 = H_3(event)$ imply $\Pr[\mathbf{E}_1 = \mathbf{E}_2] = 1$.

IV. SECURITY ANALYSIS

Theorem 7 (Anonymity)**.** *Our identity-based Linkable ring signature scheme is anonymous assuming the randomness (these hash functions are considered as random oracles), the collision-resistance of hash functions H_1, H_2, H_3.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087808, IEEE Access

Proof. We proceed the proof with a sequence of hybrid games. We will prove that these game are indistinguishable against the Anonymity adversary. We show that in the last game, the advantage of the adversary is zero. Let W_i be the event that the Anonymity adversary wins Game i.

- **Game 0.** This is the original Anonymity game via Definition 1.
- **Game 1.** Compared to Game 0, in this game we make some changes in extracting the private key for signers of id_i. Namely, once getting an extract query EQ(id_i), the challenger chooses $\mathbf{Q}_i \xleftarrow{\$} \mathbb{Z}_q^{n\times nk}$ and then programs $H_1(id_i) \leftarrow \mathbf{Q}_i$. After that, the challenger sets $\mathbf{A}_i := [\mathbf{A}|\mathbf{Q}_i]$ and then does the same steps 2-4 as in IdLRS.Extract. Note that, the challenger has to keep a list of $\mathbf{Q}_i$'s to respond consistently. In ROM, the adversary cannot detect the change between Game 1 and Game 0. Thus, we have
$$\Pr[W_1] = \Pr[W_0].$$
- **Game 2.** This game is as same as Game 4, except that when signing (in responding signing queries and in the challenge phase), the challenger uses IdLRS.Sign1 (see Figure 1 (a)). In ROM, the adversary cannot distinguish IdlRS.Sign from IdLRS.Sign1, hence
$$\Pr[W_2] = \Pr[W_1].$$
- **Game 3.** This game is as same as Game 2, except that when signing (in responding signing queries and in the challenge phase), the challenger uses IdLRS.Sign2 (see Figure 1 (b)). Lemma 6 ensures that the adversary cannot distinguish IdlRS.Sign2 from IdLRS.Sign1. Again, we have
$$\Pr[W_3] = \Pr[W_2].$$
- **Game 4.** This game is as same as Game 3, except that when signing (in responding signing queries and in the challenge phase), the challenger uses IdLRS.Sign3 (see Figure 1 (c)). In IdLRS.Sign2, a standard leftover hash lemma argument claims that $\mathbf{E} = \mathbf{K}\mathbf{S}_s$ looks like uniform. In addition, the adversary is not aware of $\mathbf{S}_s$. Then we can replace $\mathbf{E} = \mathbf{K}\mathbf{S}_s$ with $\mathbf{E} \xleftarrow{\$} \mathbb{Z}_q^{n\times n}$ without making the adversary get noticed. Hence, we have
$$\Pr[W_4] = \Pr[W_3].$$
Obviously, in this game, all $\mathbf{Q}_i$, $\mathbf{c}_i$ and all $\mathbf{z}_i$ are chosen uniformly at random from the corresponding domain. Moreover, $\mathbf{E}$ is also randomly sampled without using the secret key of the real signer (see Step 1 of IdLRS.Sign3). Therefore, the signature generated in the challenge phase is perfectly independent of choosing the signer id_{s_b}. Hence,
$$\Pr[W_5] = 0.$$

We can conclude $\Pr[W_0] = 0$. □

Theorem 8 (Unforgeability). *Our identity-based Linkable ring signature scheme satisfies Unforgeability, assuming the hardness of* $\mathsf{SIS}_{n,m+nk,q,2\Delta'}$, *where* $\Delta' := \eta\sigma\sqrt{m+nk}$, $1.1 \le \eta \le 1.3$.

Proof. To prove the proposed scheme to be secure against any existential forger, we show that if there exists a forger $\mathcal{F}$ who can compromise the unforgeability then we can construct a solver $\mathcal{S}$ being able to solve a given SIS instance. In the simulation, the real signing algorithm IdLRS.Sign is replaced with IdLRS.Sign1 and IdLRS.Sign2 (see Figure 1 (a)-(b)). The first change in both IdLRS.Sign1 and IdLRS.Sign2 compared to IdLRS.Sign is that the H_2-oracle responses are taken as a tuple of ℓ first unused values $\mathbf{c}_1, \cdots, \mathbf{c}_\ell$ from C_{H_2} (see Step 5). These H_2-oracle responses are then programmed such that for $i \in [\ell]$, $H_2(\widehat{\mathbf{A}}_i\mathbf{z}_i + q\mathbf{c}_i \bmod 2q, \widehat{\mathbf{K}}\mathbf{z}_i + q\mathbf{c}_i \bmod 2q, \widehat{\mathbf{K}} \bmod 2q, \mathcal{R}, \mu, event, \mathbf{E}) := \mathbf{c}_{i+1}$ (see Step 8). We emphasize that in the ROM setting, the forger $\mathcal{F}$ cannot distinguish between IdLRS.Sign and IdLRS.Sign1 as well as between IdLRS.Sign1 and IdLRS.Sign2 (thanks to the rejection sampling). The algorithms IdLRS.Sign1 and IdLRS.Sign2 are described as in Figure 1.

Now, suppose that there is a forger $\mathcal{F}$ that is able to break the unforgeability of the proposed scheme. Using $\mathcal{F}$, we construct an SIS solver $\mathcal{S}$ as follows:

- **SIS instance.** The SIS solver $\mathcal{S}$ is given the SIS instance $\mathbf{F}\mathbf{x} = 0 \pmod q$, $\|\mathbf{x}\| \le \beta$, $\beta = 2\Delta'$, where $\Delta' := \eta\sigma\sqrt{m+nk}$, $1.1 \le \eta \le 1.3$, $\mathbf{F} = [\mathbf{A}|\mathbf{F}_\theta]$ with $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{F}_\theta \xleftarrow{\$} \mathbb{Z}_q^{n\times nk}$.
- **Setup.** $\mathcal{S}$ first samples $\mathbf{F}_2, \cdots, \mathbf{F}_\ell \xleftarrow{\$} \mathbb{Z}_q^{n\times nk}$ then $\mathcal{S}$ selects three hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_{2q}^{n\times nk}$, $H_2 : \{0,1\}^* \to S_w^n$ and $H_3 : \{0,1\}^* \to \mathbb{Z}_{2q}^{n\times(m+nk)}$. Suppose that $\mathcal{F}$ will makes at most q_E extract queries, q_S sign queries. Note that, each sign query calls ℓ queries to the H_2 oracle. Let $q_T := q_E + \ell \cdot q_S$. In order to prepare for replying queries made by $\mathcal{F}$, $\mathcal{S}$ creates a H_1-list $\mathsf{L}_1 = \{(id_i, \mathbf{Q}_i, \mathbf{R}_i, \mathbf{A}_i, \mathsf{flag}) : H_1(id_i) := \mathbf{Q}_i, \mathbf{A}_i = [\mathbf{A}|\mathbf{Q}_i]\}$, where $\mathsf{flag} = 1$ if $\mathbf{Q}_i$ is of the form $\mathbf{Q}_i = \mathbf{G} - \mathbf{A}\mathbf{R}_i$, $\mathsf{flag} = 2$ if $\mathbf{Q}_i$ is some $\mathbf{F}_j, j \in \{2, \cdots, \ell\}$, while $\mathsf{flag} = 3$ if $\mathbf{Q}_i = \mathbf{F}_\theta$. Also, $\mathcal{S}$ creates a H_2-list L_2 consisting of tuples $((\mathbf{u}_1, \mathbf{u}_2, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E}), \mathbf{c})$ satisfying that $H_2(\mathbf{u}_1, \mathbf{u}_2, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E}) := \mathbf{c}$. $\mathcal{B}$ also selects randomly from S_w^n a set $\mathsf{C}_{H_2} := \{\mathbf{c}^{(1)}, \cdots, \mathbf{c}^{(q_T)}\}$. Additionally, $\mathcal{S}$ prepares a list L_3 of tuples $(id_i, \mathbf{S}_i)$. Moreover, $\mathcal{S}$ prepares a list L_4 of tuples $(id_i, \{\mathbf{z}_j\}_{j=1}^{\ell}, \mathcal{R}, \mathbf{c}_1, event, \mu, \mathbf{E})$ for replying sign queries. These $\mathsf{L}_1, \mathsf{L}_2, \mathsf{L}_3$ and L_4 are initially empty. The public key $pk = \mathbf{A}$ and , H_1, H_2, H_3 are sent to $\mathcal{F}$.
- **Query.**
  - H_1 *query.* Once $\mathcal{F}$ submits an identity id_i, $\mathcal{S}$ first checks whether id_i exists in the list L_1 or not. If not, $\mathcal{S}$ samples an $\mathbf{R}_i \sim D_{\sigma_2}^{n\times nk}$, simultaneously selects an unused $\mathbf{F}_{\mathsf{uns}}$ from $\{\mathbf{F}_\theta, \mathbf{F}_2, \cdots, \mathbf{F}_\ell\}$, then returns $\mathbf{Q}_i$ as follows: (i) $\mathbf{Q}_i := \mathbf{G} - \mathbf{A}\mathbf{R}_i$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087808, IEEE Access

IdLRS.Sign1($\mu, event, \mathcal{R}, sk_s$):

1) $\mathbf{K} = H_3(event)$, $\mathbf{E} = \mathbf{K}\mathbf{S}_s$.
2) $\widehat{\mathbf{K}} \leftarrow [2\mathbf{K}|-2\mathbf{E}+q\mathbf{I}_n]$, $\widehat{\mathbf{S}}_s \leftarrow \begin{bmatrix} \mathbf{S}_s \\ \mathbf{I}_n \end{bmatrix}$.
3) $\widehat{\mathbf{A}}_i \leftarrow [\mathbf{A}_i|\mathbf{0}]$, $\forall i \in [\ell]$.
4) $\mathbf{y} \leftarrow D_\sigma^{m+nk+n}$.
5) $\boxed{\mathbf{c}_1, \cdots, \mathbf{c}_\ell \xleftarrow{\$} \mathsf{C}_{H_2}}$.
6) For $j \neq s$, $\mathbf{z}_j \leftarrow D_\sigma^{m+nk+n}$.
7) For $j = s$: $b \in \{0,1\}$, output $\mathbf{z}_s \leftarrow (-1)^b \widehat{\mathbf{S}}_s \mathbf{c}_s + \mathbf{y}$ with probability $\min\left\{\frac{D_\sigma^{m+nk+n}(\mathbf{z}_s)}{M \cdot D^{m+nk+n}_{(-1)^b\widehat{\mathbf{S}}_s\mathbf{c}_s,\sigma}(\mathbf{z}_s)}, 1\right\}$.
8) Program $H_2(\widehat{\mathbf{A}}_i\mathbf{z}_i + q\mathbf{c}_i, \widehat{\mathbf{K}}\mathbf{z}_i + q\mathbf{c}_i, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E}) := \mathbf{c}_{i+1}$, $\forall i \in [\ell]$.
9) Output $\sigma_\mathcal{R} = \sigma_\mathcal{R}(\mu, event) = \big(\{\mathbf{z}_j\}_{j\in[\ell]}, \mathbf{c}_1, \mathcal{R}, \mathbf{E}\big)$.

(a)

IdLRS.Sign2($\mu, event, \mathcal{R}, sk_s$):

1) $\mathbf{K} = H_3(event)$, $\mathbf{E} = \mathbf{K}\mathbf{S}_s$.
2) $\widehat{\mathbf{K}} \leftarrow [2\mathbf{K}| - 2\mathbf{E} + q\mathbf{I}_n]$.
3) $\widehat{\mathbf{A}}_i \leftarrow [\mathbf{A}_i|\mathbf{0}]$, $\forall i \in [\ell]$.
4) $\mathbf{y} \leftarrow D_\sigma^{m+nk+n}$.
5) $\mathbf{c}_1, \cdots, \mathbf{c}_\ell \xleftarrow{\$} \mathsf{C}_{H_2}$.
6) For $j \neq s$: $\mathbf{z}_j \leftarrow D_\sigma^{m+nk+n}$.
7) For $j = s$: $\boxed{\mathbf{z}_s \leftarrow D_\sigma^{m+nk+n}}$, $\boxed{\text{output } \mathbf{z}_s \text{ with probability } 1/M}$.
8) Program $H_2(\widehat{\mathbf{A}}_i\mathbf{z}_i + q\mathbf{c}_i, \widehat{\mathbf{K}}\mathbf{z}_i + q\mathbf{c}_i, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E}) := \mathbf{c}_{i+1}$, $\forall i \in [\ell]$.
9) Output $\sigma_\mathcal{R} = \sigma_\mathcal{R}(\mu, event) = \big(\{\mathbf{z}_j\}_{j\in[\ell]}, \mathbf{c}_1, \mathcal{R}, \mathbf{E}\big)$.

(b)

IdLRS.Sign3($\mu, event, \mathcal{R}$):

1) $\mathbf{K} = H_3(event)$, $\boxed{\mathbf{E} \xleftarrow{\$} \mathbb{Z}_q^{n\times n}}$.
2) $\widehat{\mathbf{K}} \leftarrow [2\mathbf{K}| - 2\mathbf{E} + q\mathbf{I}_n]$.
3) $\widehat{\mathbf{A}}_i \leftarrow [\mathbf{A}_i|\mathbf{0}]$ $\forall i \in [\ell]$.
4) ~~$\mathbf{y} \leftarrow D_\sigma^{m+nk+n}$~~.
5) $\mathbf{c}_1, \cdots, \mathbf{c}_\ell \xleftarrow{\$} \mathsf{C}_{H_2}$.
6) For $j \neq s$: $\mathbf{z}_j \leftarrow D_\sigma^{m+nk+n}$.
7) For $j = s$: $\mathbf{z}_s \leftarrow D_\sigma^{m+nk+n}$, output $\mathbf{z}_s$ with probability $1/M$.
8) Program $H_2(\widehat{\mathbf{A}}_i\mathbf{z}_i + q\mathbf{c}_i, \widehat{\mathbf{K}}\mathbf{z}_i + q\mathbf{c}_i, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, \mathbf{E}) := \mathbf{c}_{i+1}$, $\forall i \in [\ell]$.
9) Output $\sigma_\mathcal{R} = \sigma_\mathcal{R}(\mu, event) = \big(\{\mathbf{z}_j\}_{j\in[\ell]}, \mathbf{c}_1, \mathcal{R}, \mathbf{E}\big)$.

(c)

FIGURE 1: A summary of hybrid signing algorithms. Note that, IdLRS.Sign3 can work without sk_s

with probability of $\delta = \frac{q_E}{\ell+q_E}$, sets $\mathsf{flag} = 1$; (ii) $\mathbf{Q}_i := \mathbf{F}_{\mathsf{uns}}$ with probability of $1-\delta$. If $\mathbf{F}_{\mathsf{uns}} \neq \mathbf{F}_\theta$ then $\mathcal{S}$ sets $\mathsf{flag} = 2$, and $\mathbf{R}_i = \perp$. However if $\mathbf{F}_{\mathsf{uns}} = \mathbf{F}_\theta$ then $\mathcal{S}$ sets $\mathsf{flag} = 3$, and $\mathbf{R}_i = \perp$. Afterwards, $\mathcal{S}$ then sets $\mathbf{A}_i := [\mathbf{A}|\mathbf{Q}_i]$. Note that, $\mathcal{S}$ also stores the new tuple $(id_i, \mathbf{Q}_i, \mathbf{R}_i, \mathbf{A}_i, \mathsf{flag})$ in the list L_1.

-- *H_2 query.* Once $\mathcal{F}$ submits a tuple $\mathbf{w} := (\mathbf{u}_1 \bmod 2q, \mathbf{u}_2 \bmod 2q, \widehat{\mathbf{K}} \bmod 2q, \mathcal{R}, \mu, event, \mathbf{E})$, $\mathcal{S}$ first checks whether $\mathbf{w}$ exists in the list L_2 or not. If not, $\mathcal{S}$ chooses the first unused $\mathbf{c}^{(j)}$ from C_{H_2} and programs $H_2(\mathbf{w}) := \mathbf{c}^{(j)}$. Also, $\mathcal{S}$ puts the new tuple $((\mathbf{u}_1 \bmod 2q, \mathbf{u}_2 \bmod 2q, \widehat{\mathbf{K}} \bmod 2q, \mathcal{R}, \mu, event, \mathbf{E}), \mathbf{c}^{(j)})$ into the list L_2.
-- *Extract query* EQ(id_i). $\mathcal{S}$ first checks whether id_i belongs to L_3 or not. If yes, $\mathcal{S}$ returns the corresponding $\mathbf{S}_i$. Otherwise, $\mathcal{S}$ checks whether id_i is in L_1 or not. If id_i exists in L_1 and its corresponding flag is $\mathsf{flag} \neq 1$, $\mathcal{S}$ rejects the query (with probability less than $1 - \delta^{q_E}$). If id_i exists in L_1 and its corresponding flag is $\mathsf{flag} = 1$, $\mathcal{S}$ takes $\mathbf{R}_i$. Otherwise, $\mathcal{S}$ samples an $\mathbf{R}_i \sim D_{\sigma_2}^{n\times nk}$, sets $\mathbf{Q}_i := \mathbf{G} - \mathbf{A}\mathbf{R}_i$ and $\mathbf{A}_i := [\mathbf{A}|\mathbf{Q}_i]$. (Note that, $\mathcal{S}$ also stores the new tuple $(id_i, \mathbf{Q}_i, \mathbf{R}_i, \mathbf{A}_i, \mathsf{flag} = 1)$ in L_1.) Having $\mathbf{R}_i$ already, $\mathcal{S}$ uses $\mathbf{T}_\mathbf{G}$ (a short basis for $\Lambda_q^\perp(\mathbf{G})$ mentioned in Subsection II-C) to find a short matrix $\mathbf{S}_i \in \mathbb{Z}_q^{(m+nk)\times n}$ satisfying that $\mathbf{A}_i\mathbf{S}_i = q\mathbf{I}_n \pmod{2q}$. Finally, $\mathcal{S}$ sends $\mathbf{S}_i$ to $\mathcal{F}$ as the response for the extract query on the identity id_i.
-- *Sign query* SQ($\mu, event, \mathcal{R}, id_s$). $\mathcal{S}$ first checks whether $(id_s, \mathcal{R}, event, \mu)$ in the list L_4 or not. If yes, it just returns $(id_s, \{\mathbf{z}_j\}_{j=1}^\ell, \mathcal{R}, \mathbf{c}, event, \mu, \mathbf{E})$ stored in L_4. Otherwise, $\mathcal{S}$ does the same as in H_1 *query*, H_2 *query* and *Extract query* using the lists in L_1, L_2 and L_3 for programming the values of $H_1(\cdot)$, $H_2(\cdot)$ and for returning the private keys $\mathbf{S}_i$. However, remark that if id_s has $\mathsf{flag} \neq 1$, then $\mathcal{S}$ rejects the sign query. Finally, $\mathcal{S}$ follows the algorithm IdLRS.Sign2($\mu, event, \mathcal{R}, sk_s$)), where $sk_s = \mathbf{S}_s$ and then forwards the output to $\mathcal{F}$. Also, $\mathcal{S}$ stores $(id_s, \{\mathbf{z}_j\}_{j=1}^\ell, \mathcal{R}, \mathbf{c}_1, event, \mu, \mathbf{E})$ in $\mathbf{L}_4$. During each signing process, the answers to H_2 queries are also placed in the list $\mathsf{L}_5 = \{(\mathbf{c}_1^{(j)}, ..., \mathbf{c}_\ell^{(j)}) : j \in [q_S]\}$ for the case of replying the same sign queries if necessary.

- **Forge.** With probability δ, the forger $\mathcal{F}$ outputs a ring signature $\sigma_{\mathcal{R}^*} = \sigma_{\mathcal{R}^*}(\mu^*, event^*) = \big(\{\mathbf{z}_j^*\}_{j\in[\ell]}, \mathbf{c}_1^*, \mathcal{R}^*, \mathbf{E}^*\big)$, such that SQ($\mu^*, event^*, \mathcal{R}^*, id$), $\forall id \in \mathcal{R}^*$ and extract queries EQ(id), $\forall id \in \mathcal{R}^*$ have never been made in the **Query** phase, and that
  1) For all $j \in [\ell], \|\mathbf{z}_j^*\| \leq \Delta := \eta\sigma\sqrt{m+nk+n}$ where $1.1 \leq \eta \leq 1.3$.
  2) Let $\mathbf{K} = H_3(event^*)$, and let $\widehat{\mathbf{K}} \leftarrow [2\mathbf{K}| - 2\mathbf{E}^* + q\mathbf{I}_n] \in \mathbb{Z}_{2q}^{n\times(m+nk+n)}$.
  3) For $i \in [\ell]$, do:
     -- Let $\widehat{\mathbf{A}}_i \leftarrow [\mathbf{A}_i|\mathbf{0}] \in \mathbb{Z}_{2q}^{n\times(m+nk+n)}$,
     -- Assign $\mathbf{c}_{i+1}^* \leftarrow H_2(\widehat{\mathbf{A}}_i\mathbf{z}_i^* + q\mathbf{c}_i^* \bmod 2q, \widehat{\mathbf{K}}\mathbf{z}_i^* + q\mathbf{c}_i^* \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}^*, \mu^*, event^*, \mathbf{E}^*)$.
  4) $\mathbf{c}_1^* = H_2(\widehat{\mathbf{A}}_\ell\mathbf{z}_\ell^* + q\mathbf{c}_\ell^* \bmod 2q, \widehat{\mathbf{K}}\mathbf{z}_\ell^* + q\mathbf{c}_\ell^* \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}^*, \mu^*, event^*, \mathbf{E}^*)$.
- **Analysis.** If all $id_i \in \mathcal{R}^*$ have $\mathsf{flag} \neq 3$, then $\mathcal{S}$ aborts. Otherwise, suppose that for some $s \in [\ell]$, $id_s \in \mathcal{R}^*$ has $\mathsf{flag} = 3$, i.e., $\widehat{\mathbf{A}}_s = [\mathbf{A}|\mathbf{F}_\theta|\mathbf{0}]$.
  Notice that, if the sign query SQ($\mu^*, event^*, \mathcal{R}^*, id_s$)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087808, IEEE Access

was not made as well as the random oracle H_2 was not called or programmed on input $(\widehat{\mathbf{A}}_s\mathbf{z}_s^* + q\mathbf{c}_s^*, \widehat{\mathbf{K}}\mathbf{z}_s^* + q\mathbf{c}_s^*, \widehat{\mathbf{K}}, \mathcal{R}^*, \mu^*, event^*, \mathbf{E}^*)$, then $\mathcal{F}$ has $\frac{1}{|S_w^n|}$ chances of producing a $\mathbf{c}_{s+1 \bmod \ell}^*$ such that $\mathbf{c}_{s+1 \bmod \ell}^* = H_2(\widehat{\mathbf{A}}_s\mathbf{z}_s^* + q\mathbf{c}_s^*, \widehat{\mathbf{K}}\mathbf{z}_s^* + q\mathbf{c}_s^*, \widehat{\mathbf{K}}, \mathcal{R}^*, \mu^*, event^*, \mathbf{E}^*)$. This turns out that $\mathbf{c}_{s+1 \bmod \ell}^* = \mathbf{c}^{(j)} \in \mathsf{C}_{H_2}$ for some $j \in [q_T]$ with probability $\delta\left(1 - \frac{1}{|S_w^n|}\right) \geq \delta - \frac{1}{|S_w^n|}$.

At this point, the solver $\mathcal{S}$ runs again the attack above of $\mathcal{F}$ but this time with $\mathsf{C}'_{H_2} := \{\mathbf{c}^{(1)}, \cdots, \mathbf{c}^{(j-1)}, \mathbf{c}'^{(j)}, \cdots, \mathbf{c}'^{(q_T)}\}$ instead of C'_{H_2} on the same message μ^*, the same $event^*$ and the same ring $\mathcal{R}^*$, in which $\mathbf{c}'^{(j)}, \cdots, \mathbf{c}'^{(q_T)}$ are new freshly chosen from S_w^n. The forking lemma [8] says that the probability that $\mathbf{c}^{(j)} \neq \mathbf{c}'^{(j)}$ and that the forger $\mathcal{F}$ uses $\mathbf{c}'^{(j)}$ in his forgery is not smaller than

$$\left(\delta - \frac{1}{|S_w^n|}\right)\left(\frac{\delta - \frac{1}{|S_w^n|}}{q_T} - \frac{1}{|S_w^n|}\right). \quad (1)$$

With the probability (1), $\mathcal{F}$ forges a new signature $\sigma'_{\mathcal{R}^*} = \sigma'_{\mathcal{R}^*}(\mu^*, event^*) = (\{\mathbf{z}_j'^*\}_{j\in[\ell]}, \mathbf{c}_1'^*, \mathcal{R}^*, \mathbf{E}^*)$, where $\mathbf{c}_1'^* = \mathbf{c}'^{(j)}$, $\widehat{\mathbf{A}}_s\mathbf{z}_s^* + q\mathbf{c}^{(j)} = \widehat{\mathbf{A}}_s\mathbf{z}_s'^* + q\mathbf{c}'^{(j)}$ and $\widehat{\mathbf{K}}\mathbf{z}_s^* + q\mathbf{c}^{(j)} = \widehat{\mathbf{K}}\mathbf{z}_s'^* + q\mathbf{c}'^{(j)}$. This implies that

$$\widehat{\mathbf{A}}_s\left(\mathbf{z}_s^* - \mathbf{z}_s'^*\right) = q(\mathbf{c}'^{(j)} - \mathbf{c}^{(j)}) \bmod 2q,$$

which is equivalent to $[\mathbf{A}|\mathbf{F}_\theta|\mathbf{0}]\left(\mathbf{z}_s^* - \mathbf{z}_s'^*\right) = q(\mathbf{c}'^{(j)} - \mathbf{c}^{(j)}) \bmod 2q$.

Separate $\mathbf{z}_s^* - \mathbf{z}_s'^*$ as $\begin{pmatrix}\mathbf{z}_{1,s}^* - \mathbf{z}_{1,s}'^* \\ \mathbf{z}_{2,s}^* - \mathbf{z}_{2,s}'^*\end{pmatrix}$, where $\mathbf{z}_{1,s}^* - \mathbf{z}_{1,s}'^* \in \mathbb{Z}_{2q}^{m+nk}$, $\mathbf{z}_{2,s}^* - \mathbf{z}_{2,s}'^* \in \mathbb{Z}_{2q}^n$.

Let $\widehat{\mathbf{x}} := \mathbf{z}_{1,s}^* - \mathbf{z}_{1,s}'^*$. Notice that since $\mathbf{c}'^{(j)} - \mathbf{c}^{(j)} \neq 0 \bmod 2$, then we have $\widehat{\mathbf{x}} \neq 0 \bmod 2q$, where $\|\widehat{\mathbf{x}}\| \leq 2\Delta' < q$. We thus have $\widehat{\mathbf{x}} \neq 0 \bmod q$ and $[\mathbf{A}|\mathbf{F}_\theta]\widehat{\mathbf{x}} = 0 \pmod q$. . This implies that $\mathbf{F}\mathbf{x} = 0 \pmod q$, where $\mathbf{x}$ is obtained from $\widehat{\mathbf{x}}$ by inserting zero rows into appropriate positions. Therefore, $\mathcal{S}$ obtains a solution of the given SIS problem.

□

Theorem 9 (Linkability)**.** *Our identity-based Linkable ring signature scheme satisfies Linkability, assuming the hardness of* $\mathsf{SIS}_{n,m+nk,q,2\Delta'}$*, where* $\Delta' := \eta\sigma\sqrt{m+nk}$*,* $1.1 \leq \eta \leq 1.3$.

Proof. The proof is quite similar to that of Theorem 8. Suppose that there is an attacker $\mathcal{A}$ that is able to break the linkability of the scheme. Using $\mathcal{A}$, we construct an SIS solver $\mathcal{B}$ as follows:

- **SIS instance.** The SIS solver $\mathcal{B}$ is given the SIS instance $\mathbf{F}\mathbf{x} = 0 \pmod q$, $\|\mathbf{x}\| \leq \beta$, $\beta = 2\Delta'$, where $\Delta' := \eta\sigma\sqrt{m+nk}$, $1.1 \leq \eta \leq 1.3$, $\mathbf{F} = [\mathbf{A}|\mathbf{F}_\theta]$ with $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{F}_\theta \xleftarrow{\$} \mathbb{Z}_q^{n\times nk}$.
- **Setup.** $\mathcal{B}$ guesses the real signer of identity, say id_π, that $\mathcal{A}$ wants to attack. The rests are the same as the **Setup** phase in the proof of Theorem 8.
- **Query.** Same as the **Query** phase in the proof of Theorem 8, except that $H_1(id_\pi) = \mathbf{G} - \mathbf{A}\mathbf{R}_\pi$, for some $\mathbf{R}_\pi \sim D_{\sigma_2}^{n\times nk}$, $\mathbf{A}_\pi := [\mathbf{A}|\mathbf{G} - \mathbf{A}\mathbf{R}_\pi]$ and its flag $= 1$.
- **Unlink.** Eventually, $\mathcal{A}$ outputs $\sigma_{\mathcal{R}}^{(1)} = \sigma_{\mathcal{R}}^{(1)}(\mu, event^*) = \left(\{\mathbf{z}_j^{(1)}\}_{j\in[\ell]}, \mathbf{c}_1^{(1)}, \mathcal{R}, \mathbf{E}\right)$ and $(\sigma_{\mathcal{R}'} = \sigma_{\mathcal{R}'}(\mu', event^*) = \left(\{\mathbf{z}'_j\}_{j\in[\ell]}, \mathbf{c}'_1, \mathcal{R}', \mathbf{E}'\right)$ such that
  1) For all $j \in [\ell], \|\mathbf{z}_j^{(1)}\|, \|\mathbf{z}'_j\| \leq \Delta := \eta\sigma\sqrt{m+nk+n}$ where $1.1 \leq \eta \leq 1.3$.
  2) Let $\mathbf{K} = H_3(event^*)$, and let $\widehat{\mathbf{K}} \leftarrow [2\mathbf{K}| - 2\mathbf{E} + q\mathbf{I}_n] \in \mathbb{Z}_{2q}^{n\times(m+nk+n)}$, $\widehat{\mathbf{K}}' \leftarrow [2\mathbf{K}| - 2\mathbf{E}' + q\mathbf{I}_n] \in \mathbb{Z}_{2q}^{n\times(m+nk+n)}$.
  3) For $i \in [\ell]$, if we let
     - $\widehat{\mathbf{A}}_i \leftarrow [\mathbf{A}_i|\mathbf{0}] \in \mathbb{Z}_{2q}^{n\times(m+nk+n)}$,
     - $\mathbf{c}_{i+1 \bmod \ell}^{(1)} \leftarrow H_2(\widehat{\mathbf{A}}_i\mathbf{z}_i^{(1)} + q\mathbf{c}_i^{(1)}, \widehat{\mathbf{K}}\mathbf{z}_i^{(1)} + q\mathbf{c}_i^{(1)}, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event^*, \mathbf{E})$, and
     - $\mathbf{c}'_{i+1 \bmod \ell} \leftarrow H_2(\widehat{\mathbf{A}}_i\mathbf{z}'_i + q\mathbf{c}'_i, \widehat{\mathbf{K}}\mathbf{z}'_i + q\mathbf{c}'_i, \widehat{\mathbf{K}}, \mathcal{R}', \mu', event^*, \mathbf{E}')$,

     then we have
     - $\mathbf{c}_1^{(1)} = H_2(\widehat{\mathbf{A}}_\ell\mathbf{z}_\ell^{(1)} + q\mathbf{c}_\ell^{(1)}, \widehat{\mathbf{K}}\mathbf{z}_\ell^{(1)} + q\mathbf{c}_\ell^{(1)}, \widehat{\mathbf{K}}, \mathcal{R}, \mu^{(1)}, event^*, \mathbf{E})$, and
     - $\mathbf{c}'_1 = H_2(\widehat{\mathbf{A}}_\ell\mathbf{z}'_\ell + q\mathbf{c}'_\ell, \widehat{\mathbf{K}}'\mathbf{z}'_\ell + q\mathbf{c}'_\ell, \widehat{\mathbf{K}}', \mathcal{R}', \mu', event^*, \mathbf{E}')$.
  4) $\mathsf{IdLRS.Link}(\sigma_{\mathcal{R}}^{(1)}, \sigma'_{\mathcal{R}}) = \mathsf{unlinked}$, i.e., $\mathbf{E} \neq \mathbf{E}'$.

  Remind that, by the guess of $\mathcal{B}$, $\mathcal{A}$ behaves as the real signer id_π who was given the corresponding private key $\mathbf{S}_\pi$, and that $id_\pi \in \mathcal{R} \cap \mathcal{R}'$ as otherwise then $\mathcal{B}$ can abort and restart the simulation. Also, if all $id_i \in \mathcal{R}\cup\mathcal{R}'$ have $\mathsf{flag} \neq 3$, then $\mathcal{S}$ aborts. Otherwise, suppose that for some $s \in [\ell]$, $id_s \in \mathcal{R}$ has $\mathsf{flag} = 3$, i.e., $\widehat{\mathbf{A}}_s = [\mathbf{A}|\mathbf{F}_\theta|\mathbf{0}]$.
- **Analysis.** $\mathcal{B}$ computes $\mathbf{E}_\pi = \mathbf{K}\mathbf{S}_\pi$, where $\mathbf{K} = H_3(event)$. Then one of $\mathbf{E}$ and $\mathbf{E}'$ has to be different from $\mathbf{E}_\pi$. Without loss of generality, we assume that $\mathbf{E}_\pi \neq \mathbf{E}$. Same as the proof of Theorem 8, if the sign query $\mathrm{SQ}(\mu, event^*, \mathcal{R}, id_\pi)$ was not made as well as the random oracle H_2 was not called or programmed on input $(\widehat{\mathbf{A}}_s\mathbf{z}_s^{(1)} + q\mathbf{c}_s^{(1)}, \widehat{\mathbf{K}}\mathbf{z}_s^{(1)} + q\mathbf{c}_s^{(1)}, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event^*, \mathbf{E})$, then $\mathcal{F}$ has $\frac{1}{|S_w^n|}$ chances of producing a $\mathbf{c}_{s+1}^{(1)}$ such that $\mathbf{c}_{s+1}^* = H_2(\widehat{\mathbf{A}}_s\mathbf{z}_s^{(1)} + q\mathbf{c}_s^{(1)}, \widehat{\mathbf{K}}\mathbf{z}_s^{(1)} + q\mathbf{c}_s^{(1)}, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event^*, \mathbf{E})$. This turns out that $\mathbf{c}_{s+1}^{(1)} = \mathbf{c}^{(j)} \in \mathsf{C}_{H_2} := \{\mathbf{c}^{(1)}, \cdots, \mathbf{c}^{(q_T)}\}$ for some $j \in [q_T]$ with probability $\delta\left(1 - \frac{1}{|S_w^n|}\right) \geq \delta - \frac{1}{|S_w^n|}$.

  Now, by creating new lists $\mathsf{L}_2^{(2)}$ and $\mathsf{C}_{H_2}^{(2)} := \{\mathbf{c}^{(1)}, \cdots, \mathbf{c}^{(j-1)}, \mathbf{c}'^{(j)}, \cdots, \mathbf{c}'^{(q_T)}\}$ for H_2 queries, in which $\mathbf{c}'^{(j)}, \cdots, \mathbf{c}'^{(q_T)}$ are new freshly chosen from S_w^n, and following the forking lemma [8], $\mathcal{B}$ rewinds the attack by $\mathcal{A}$. Note that, at this time, H_2 queries are responded with the lists $\mathsf{L}_2^{(2)}$ and $\mathsf{C}_{H_2}^{(2)}$ instead of L_2

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087808, IEEE Access

and C_{H_2}. Again, $\mathcal{A}$ outputs a new valid ring signature $(\sigma^{(2)}_{\mathcal{R}} = \sigma^{(2)}_{\mathcal{R}}(\mu, event^*) = \left(\{\mathbf{z}^{(2)}_j\}_{j\in[\ell]}, \mathbf{c}^{(2)}_1, \mathcal{R}, \mathbf{E}\right)$, with $\mathbf{c}^{(2)}_{s+1} = \mathbf{c}'^{(j)}$ also with the probability (1) and $\widehat{\mathbf{A}}_s\mathbf{z}^{(1)}_s + q\mathbf{c}^{(1)}_s = \widehat{\mathbf{A}}_s\mathbf{z}^{(2)}_s + q\mathbf{c}^{(2)}_s$ and $\widehat{\mathbf{K}}\mathbf{z}^{(1)}_s + q\mathbf{c}^{(1)}_s = \widehat{\mathbf{K}}\mathbf{z}^{(2)}_s + q\mathbf{c}^{(2)}_s$.

Here $\mathbf{c}^{(2)}_{i+1 \bmod \ell} \leftarrow H_2(\widehat{\mathbf{A}}_i\mathbf{z}^{(2)}_i + q\mathbf{c}^{(2)}_i, \widehat{\mathbf{K}}\mathbf{z}^{(2)}_i + q\mathbf{c}^{(2)}_i, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event^*, \mathbf{E})$, for $i \in [\ell]$. Then, $\widehat{\mathbf{A}}_s(\mathbf{z}^{(1)}_s - \mathbf{z}^{(2)}_s) = q(\mathbf{c}^{(2)}_s - \mathbf{c}^{(1)}_s)$ and $\widehat{\mathbf{K}}(\mathbf{z}^{(1)}_s - \mathbf{z}^{(2)}_s) = q(\mathbf{c}^{(2)}_s - \mathbf{c}^{(1)}_s)$. The rest is similar to Theorem 8.

□

Theorem 10 (Nonslanderability)**.** *Our identity-based Linkable ring signature scheme satisfies Nonslanderability.*

Proof. We will show briefly that the nonslanderability of the proposed scheme is ensured by its unforgeability and linkability.

Indeed, recall that, in the nonslanderability game (GAME IV), at the **Challenge** phase, the adversary $\mathcal{N}$ submits a tuple of $(\mu, event, \mathcal{R}, id_s \in \mathcal{R})$, such that $\mathsf{EQ}(id_s)$ has not been queried before. The challenger $\mathcal{C}$ generates sk_s by $\mathsf{IdLRS.Extract}(pp, id_s, msk)$ and returns $\sigma_{\mathcal{R}} = \sigma_{\mathcal{R}}(\mu, event) = \left(\{\mathbf{z}_j\}_{j\in[\ell]}, \mathbf{c}_1, \mathcal{R}, \mathbf{E}\right) \leftarrow \mathsf{IdLRS.Sign}(pp, event, \mu, \mathcal{R}, sk_s)$. The adversary $\mathcal{N}$ outputs a new valid signature $\sigma'_{\mathcal{R}} = \sigma'_{\mathcal{R}}(\mu, event) = \left(\{\mathbf{z}'_j\}_{j\in[\ell]}, \mathbf{c}'_1, \mathcal{R}, \mathbf{E}'\right)$ on the same message μ and the same event $event$, in which $\mathbf{E}' = \mathbf{E}$. This means that $\mathcal{N}$ can create a signature with the linkability tag $\mathbf{E}$ without knowing sk_s but some sk_π with $id_\pi \in \mathcal{R} \setminus \{id_s\}$. Replaying the attack, $\mathcal{N}$ can also produce a valid $\sigma''_{\mathcal{R}} = \sigma''_{\mathcal{R}}(\mu, event) = \left(\{\mathbf{z}''_j\}_{j\in[\ell]}, \mathbf{c}''_1, \mathcal{R}, \mathbf{E}''\right)$, where $\mathbf{E}'' = \mathbf{E}'$, using the same sk_π. Two signatures $\sigma'_{\mathcal{R}}(\mu, event)$ and $\sigma''_{\mathcal{R}}(\mu, event)$ will be given to an SIS solver $\mathcal{S}$, who can extract a solution to the SIS instance presented in proof of Theorem 8.

Also. notice that, in the case that $\mathcal{N}$ can produce a valid $\sigma''_{\mathcal{R}} = \sigma''_{\mathcal{R}}(\mu, event) = (\{\mathbf{z}''_j\}_{j\in[\ell]}, \mathbf{c}''_1, \mathcal{R}, \mathbf{E}'')$, where $\mathbf{E}'' = \mathbf{E}'$, using a private key sk_i such that $id_i \neq id_s$ then by Theorem 9 guarantees that two valid signatures created by different users are $\mathsf{unlinked}$.

□

V. IMPLEMENTATION

As a proof of concept as well as in order to see how the practicability of the proposed IdLRS scheme is, we implemented the proposed IdLRS scheme and ran some experiments on it. In this section, we first show how to choose parameters in general. We then choose some concrete tuples of parameters being used in our experiments. Finally, we give the experimental results.

A. SETTING PARAMETERS

We follow [33] for setting heuristic parameters. We also take the security proof (was done in the Section IV) into account.

- For $\mathsf{GenTrap}$ to work: $q \geq 2, k = \lceil \log q \rceil, \overline{m} \geq 1, m = \overline{m} + nk \geq 2n \log q$.
- The parameter w defines the size of the challenges $\mathbf{c}_i$, in order to have the min-entropy at least λ, we should choose w to satisfy $2^w \cdot \binom{n}{w} \geq 2^\lambda$. Here λ is chosen depending on the value of n. In our experiments , we set $\lambda = n$.
- For Gaussian parameter in $\mathsf{GenTrap}$: via [33], for any ϵ, we should choose $\sigma_1 \geq \eta_\epsilon(\mathbb{Z})$, i.e., $\sigma_1 = \sqrt{\frac{\ln(2(1+1/\epsilon))}{\pi}}$. Also, we can choose $\epsilon = \mathrm{negl}(n)$ such that $\sigma_1 = \omega(\sqrt{\log n})$.
- For Gaussian parameter in $\mathbf{R}_i \in \mathbb{Z}^{m\times nk} \leftarrow \mathsf{DelTrap}(\mathbf{A}, \mathbf{Q}_i, \mathbf{R}, \sigma_2)$: By Lemma 5, $s_1(\mathbf{R}) \leq \sigma_1 \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{\overline{m}} + \sqrt{nk})$. We need $\sigma_2 \geq \eta_\epsilon(\Lambda^\perp_q(\mathbf{A}))$, i.e., $\sigma_2 \geq \sqrt{5} \cdot (s_1(\mathbf{R}) + 1) \cdot \omega(\sqrt{\log n})$.
- For Gaussian parameter in $\mathbf{S}_i \in \mathbb{Z}^{(m+nk)\times n} \leftarrow \mathsf{SampleD}(\mathbf{A}_i, \mathbf{R}_i, \mathbf{u}_t, \sigma_3)$: First, note that, by Lemma 5, $s_1(\mathbf{R}_i) \leq \sigma_2 \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{m} + \sqrt{nk})$. Now, $\sigma_3 \geq \sqrt{7 \cdot (s_1(\mathbf{R}_i)^2 + 1)} \cdot \omega(\sqrt{\log n})$.
- Gaussian parameter σ in rejection sampling: We have $\|\mathbf{S}_i\| \leq \cdot\sigma_3\sqrt{m+nk}$ with overwhelming probability. By Item 4 of Lemma 4, we choose $\sigma \geq \omega(\|(-1)^b\widehat{\mathbf{S}}_s\mathbf{c}_s\| \cdot \sqrt{\log(m+nk+n)})$. However, we can choose $\sigma \geq \omega(\|\mathbf{S}_s\| \cdot \|\mathbf{c}_s\| \cdot \sqrt{\log(m+nk+n)})$.
- For the $\mathsf{SIS}_{n,m+nk,q,\beta}$ problem in the security proof (see Section IV) to be hard and to have a solution: $q \geq \beta \cdot \omega(\sqrt{n \log n})$, $\beta \geq \sqrt{m+nk} \cdot q^{n/(m+nk)}$ where $\beta = 2\Delta' := 2\eta\sigma\sqrt{m+nk}$ where $1.1 \leq \eta \leq 1.3$. (See Remark 1 for η and see also Section IV for why $\beta = 2\Delta'$.)
- Choose M in the rejection sampling: Remark 2 claims that if $\sigma = 12\|\mathbf{c}\|$, then $\frac{D^m_\sigma(\mathbf{x})}{M \cdot D^m_{\mathbf{c},\sigma}(\mathbf{x})} \leq \frac{e^{1+1/288}}{M} \geq \frac{3}{M}$ with probability bigger than $1 - 2^{-100}$. Then we should choose $\sigma \geq \max\{\omega(\|\mathbf{S}_s\| \cdot \sqrt{\log(m+nk+n)}), 12\} \cdot \|\mathbf{c}_s\|$ then can fix $M = 3$.

Now, in order to set parameters for our experiments, we first choose n. We will choose the modulus q to be a power of two integer, i.e., $q = 2^k$ for some positive integer k. Note that, aiming to choose $m = 2nk$, from $q \geq \sqrt{m+nk} \cdot q^{n/(m+nk)} \cdot \omega(\sqrt{n\log(n)})$ we have $2k - \log(k) \geq \omega(2/3 + \log(3) + 2\log(n) + \log(\log(n)))$. Therefore, given n, we can choose k using this condition. Now we set $q = 2^k$. We choose $\overline{m} = nk$ to have $m = \overline{m} + nk = 2nk$. We choose w to be the smallest such that $2^w \cdot \binom{n}{w} \geq 2^n$.

For Gaussian parameters, we set $\sigma_1 = \eta_\epsilon(\mathbb{Z}) = \sqrt{\frac{\ln(2(1+1/\epsilon))}{\pi}} = 4.48083023712027$, where $\epsilon = 2^{-90}$, $\sigma_2 = \sqrt{5} \cdot (\sigma_1 \cdot \frac{1}{\sqrt{2\pi}} \cdot (\sqrt{\overline{m}} + \sqrt{nk}) + 1) \cdot \omega(\log(n))$, and $\sigma_3 = \sqrt{7(s_i^2 + 1)} \cdot \omega(\log(n)))$, where $s_i = \sigma_2 \cdot (1/\sqrt{2\pi}) \cdot (\sqrt{m} + \sqrt{nk})$. For the Gaussian parameter σ used in the rejection sampling, we set $\sigma = \omega(\sigma_3 \cdot \sqrt{m+nk} \cdot w \cdot \max\{12, \sqrt{\log(m+nk+n)}\})$. Finally we set $M = 3$.

Specifically, we will run experiments with the following concrete tuples of parameters:

- $pp_1 : n = 40, q = 2^{26}, \overline{m} = 1040, w = 11, m = 2080, M = 3$ and $\sigma_1 = 4.48083023712027$, $\sigma_2 =$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087808, IEEE Access

1165.22352070264, $\sigma_3 = 429061.131614986$, $\sigma = 8.11257427288557e10$

- $pp_2 : n = 60, q = 2^{29}, \overline{m} = 1740, w = 15, m = 3480, M = 3$ and $\sigma_1 = 4.48083023712027$, $\sigma_2 = 1504.24674659589$, $\sigma_3 = 716451.780933469$, $\sigma = 3.21126195212814e11$
- $pp_3 : n = 80, q = 2^{32}, \overline{m} = 2560, w = 20, m = 5120, M = 3$ and $\sigma_1 = 4.48083023712027$, $\sigma_2 = 1822.45271625383$, $\sigma_3 = 1.05285730202738e6$, $\sigma = 2.87875508121890e12$
- $pp_4 : n = 100, q = 2^{35}, \overline{m} = 3500, w = 24, m = 7000, M = 3$ and $\sigma_1 = 4.48083023712027$, $\sigma_2 = 2129.23955112039$, $\sigma_3 = 1.43830770969715e6$, $\sigma = 5.51800764686501e13$

B. EXPERIMENTAL RESULTS

We implemented the proposed IdLRS using SageMath 9.2 [2] which in turn bases on Python 3.8. The source code can be publicly accessed via Github [3]. We ran our experiments on the sever Dell Poweredge R730 installing Ubuntu 18.04.5 TLS with Memory 40GB, Processor Intel Xeon 8-core 2.1GHz. For each tuple of parameters pp_i above, we ran 5 times and computed the average times of the key generation algorithm and extraction algorithm and also the average sizes of public key, of master secret key and of private key. Having used one of the tupe of public key, master secret key and private key, we then ran 10 times the experiments with the rings having 10, 20, 30 and 40 members. By doing this, we evaluated the average times of the signing algorithm, verifying algorithm, the linking algorithm and also the average sizes of the corresponding signatures.

We summarize the running times of algorithms in our experiments in Table 3, and then visualize them together in Figures 2. Also, the sizes of public key, master secret key, private key and signatures in both theoretical estimation (entitled "Theo.“) and our experiments (entitled "Exp.“) are included in Table 4. We then plot these sizes to see how they vary in Figure 3.

It can be seen from Table 3 and Figure 2 that the extraction algorithm consumes quite much time in comparison with other algorithms. This is because the extraction algorithm (IdLRS.Ext) exploits the DelTrap which calls up to $n \cdot k$ times SampleD. (For example, with the tuple of public parameters pp_4, we have $n \cdot k = 3500$.) SampleD in turn calls the Gaussian sampling algorithm over lattices. We implemented the Gaussian sampling algorithm following the one in [19], which is quite inefficient. Moreover, our implementation is actually not optimal. Therefore, the experimental results should be much better in terms of runtime and even of size if:

- We can implement using more efficient Gaussian sampling algorithms over lattices, e.g. [39], [18] [4]. We instead implement the ring-based IdLRS version, which will be presented in Section VI below. We can optimise and parallelise the code.

[2]https://www.sagemath.org/index.html
[3]https://github.com/huyle84/identity-based-linkable-ring-signature
[4]For more updated details on Gaussian sampling, we refer readers to the link https://cseweb.ucsd.edu/ daniele/LatticeLinks/Sampling.html.

VI. A CONSTRUCTION BASED ON RING-SIS

In this section, for $n \in \mathbb{N}$ and any prime q, we consider the cyclotomic polynomial rings $R := \mathbb{Z}[x]/\langle x^n+1\rangle$ and $R_q := R/qR = \mathbb{Z}_q[x]/\langle x^n+1\rangle$. Note that there is an isomorphic berween R and $\mathbf{Z}^n$ as well as between R_q and $\mathbf{Z}_q^n$ through the canonical embedding (see [33], [32]). For a polynomial $f = f_0 + f_1x + \cdots + f_{n-1}x^{n-1} \in R$, define $\|f\|_\infty := \max|f_i|$, $\|f\| := \sum_{i=1}^{n-1} f_i^2$ and $\|f\|_1 := \sum_{i=1}^{n-1}|f_i|$. Also, for any $f \in R$ we denie $s_1(f) := \sup_{g\in R} \frac{\|f\cdot g\|}{\|g\|}$.

First, we will present the definition of ring-SIS problem, the gadget-based trapdoor in ideal lattices and its related algorithms. We then give the construction.

Definition 8 (ring-SIS Problem)**.** *Given positive integers q, m, and random vector $\mathbf{A} \xleftarrow{\$} R_q^{1\times m}$ and $\beta \in \mathbb{R}^+$, the* ring-SIS$_{m,q,\beta}$ *problem requires to seek a non-zero short vector $\mathbf{E} \in R^{m\times 1}$ satisfying $\|\mathbf{E}\|_\infty \leq \beta$ and $\mathbf{AE} = 0 \pmod q$.*

The hardness of ring-SIS$_{m,q,\beta}$ is proved in e.g., [40], [31] [30] through a reduction from the shortest vector problem (a.k.a, γ-IdealSVP) over ideal lattices. Formally, defining $\theta := \max_{g\in\mathbb{Z}[x],\deg(g)\leq 3(n-1)} \frac{\|g \bmod (x^n+1)\|_\infty}{\|g\|_\infty}$, we have

Theorem 11 ([30, Theorem 2.3 for $f = x^n + 1$])**.** *For $q > 2\theta\beta m n^{1.5}\log n$, if there is a polynomial-time algorithm that solves the* ringSIS$_{m,q,\beta}$ *problem with some non-negligible probability, then there is a polynomial-time algorithm that solves the γ-*IdealSVP *problem with $\gamma = 8\theta\beta m n \log^2 n$ for any lattice Λ that corresponds to an ideal in R.*

We exploit the trapdoor mechanism for ideal lattices which was proposed in [33] then detailed in Lai et al. [22]. It was also improved by Genise and Miciancio [18] and then summarized in Bert et al. [9].

Definition 9 ($\mathbf{g}$-Trapdoor, [18, Definition 3])**.** *Let $\mathbf{a} \in R_q^m$ and $\mathbf{g} \in R_q^k$, where $k = \lceil \log q \rceil$ be vectors of polynomials, with $m > k$. A matrix $\mathbf{R} \in R^{(m-k)\times k}$ is called $\mathbf{g}$-trapdoor for $\mathbf{a}$ with tag h (which is an invertible element in $\mathbb{R}_q$) if $\mathbf{a}^t \cdot \begin{bmatrix}\mathbf{R}\\ \mathbf{I}_k\end{bmatrix} = h\mathbf{g}^t$.*

Lemma 12 ([15, Lemma 5])**.** *For any polynomial $r := r(x) = r_0 + r_1x + \cdots r_{n-1}x^{n-1}$ in R, we have $s_1(r) \leq \|r\|_1 := \sum_{i=0}^{n-1} r_i$.*

Lemma 13 ([15, Fact 6])**.** *If $\mathbf{R} \leftarrow D_{R,\sigma}^{w\times k}$ then with overwhelming probability, we have $s_1(\mathbf{R}) \leq \sigma\sqrt{n}\cdot O(\sqrt{w} + \sqrt{k} + \omega(\sqrt{\log n}))$.*

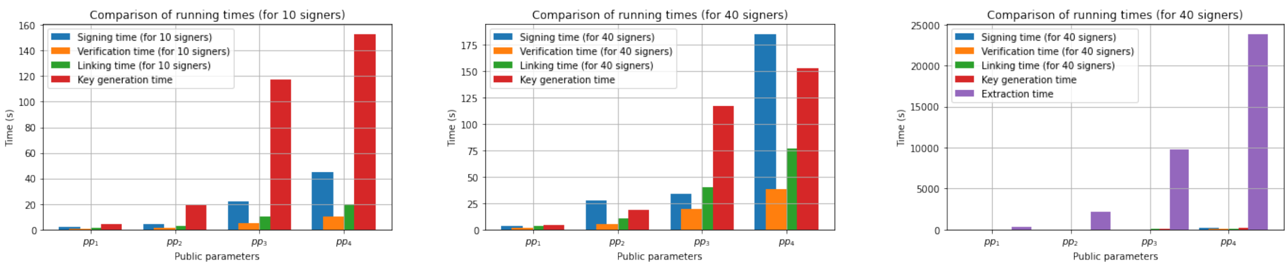
We consider a vector of constant polynomials $\mathbf{g}^t = (1, 2, 4, ..., 2^{k-1}) \in R_q^k$, where $k = \lceil \log_2 q \rceil$. We can find a publicly known short basis, say $\mathbf{B}_k \in R^{k\times k}$ for $\Lambda^\perp(\mathbf{g}^t)$, i.e., $\mathbf{g}^t.\mathbf{B}_k = \mathbf{0} \in R_q^k$ and $\|\widetilde{\mathbf{B}_k}\| \leq \sqrt{5}$.

Parameters	pp_1				pp_2				pp_3				pp_4			
Key generation time	4				19				117				153			
Extraction time	378				2115				9773				23861			
# Signers in ring	10	20	30	40	10	20	30	40	10	20	30	40	10	20	30	40
Signing time	1.9	2.5	3.1	3.7	4.6	5.5	18.6	27.4	11.9	15.3	26.4	33.8	45.1	126.9	142.1	185.4
Verifying time	0.5	1	1.5	1.9	1.4	2.7	3.9	5.3	5.1	10.1	14.1	19.9	9.9	19.2	28.7	38
Linking time	1.1	1.9	3.1	3.8	2.9	5.3	7.7	10.7	10,3	20.1	28	40	19.9	38.2	57.4	76.4

TABLE 3: Running times (in second).

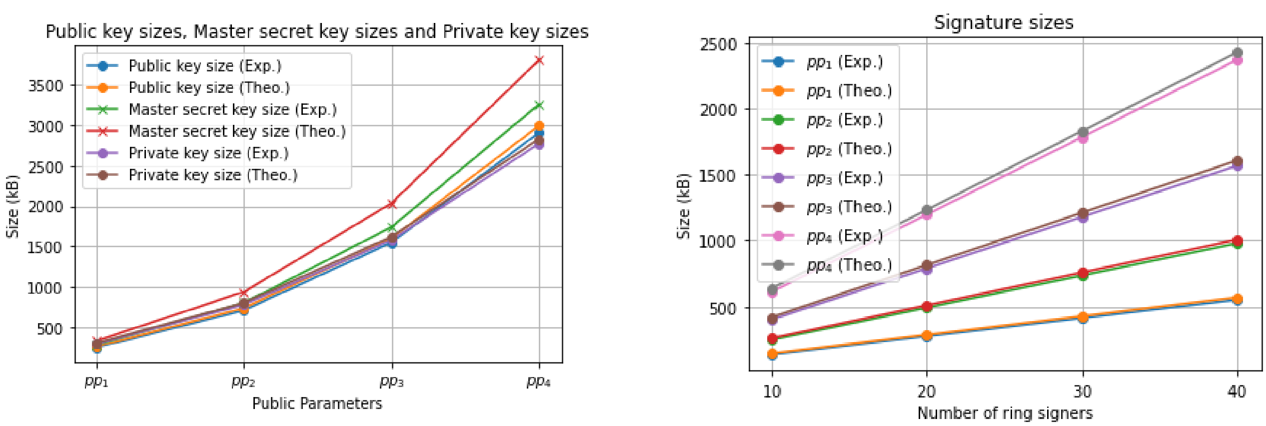
Parameters	pp_1				pp_2				pp_3				pp_4			
Public key (Theo.)	265				740				1600				2991			
Public key (Exp.)	253				713				1550				2905			
Master secret key (Theo.)	336				940				2034				3802			
Master secret key (Exp.)	287				804				1741				3254			
Private key (Theo.)	310				805				1620				2826			
Private key (Exp.)	299				782				1582				2770			
# Signers in ring	10	20	30	40	10	20	30	40	10	20	30	40	10	20	30	40
Signature (Theo.)	147	288	429	571	262	511	760	1008	421	817	1212	1608	639	1234	1830	2426
Signature (Exp.)	139	277	415	553	250	493	737	981	403	791	1180	1568	613	1198	1784	2369

TABLE 4: Sizes (in kB).



(a) Without Extraction time (for 10 signers) (b) Without Extraction time (for 40 signers) (c) With Extraction time (for 40 signers)

FIGURE 2: Comparison of running times in our experiments.



(a) Public key sizes, master secret key sizes and private key sizes (b) Signature sizes

FIGURE 3: Comparison of sizes in both the theoretical estimation (Theo.) (via Table 2) and our experiments (Exp.).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087808, IEEE Access

The following algorithms enable us to generate a vector of polynomials and its **g**-trapdoor, to sample via Gaussian and to delegate trapdoors in the ideal lattice setting.

$\mathsf{GenTrap}(m, q, \sigma, h)$:

- **Input:** q, $k = \log q$, $m > k$, and Gaussian parameter σ.
- **Ouput:** A vector $\mathbf{a} \in R_q^m$ and **g**-trapdoor $\mathbf{R}$ for $\mathbf{a}$.
- **Execute:**
  1) Choose $\bar{\mathbf{a}} \xleftarrow{\$} R_q^{m-k}$.
  2) Choose $\mathbf{R} \in R^{(m-k)\times k}$ from the distribution $D_{R^{(m-k)\times k},\sigma}$
  3) Output $\mathbf{a} = (\bar{\mathbf{a}}^t, h\mathbf{g}^t - \bar{\mathbf{a}}^t\mathbf{R})^t \in R_q^m$, trapdoor $\mathbf{R}$.

$\mathsf{SamplePre}(\mathbf{a}, \mathbf{R}, h, u, \zeta, \sigma, \alpha)$:

- **Input:** $\mathbf{a} \in R_q^m$ and its **g**-Trapdoor matrix $\mathbf{R} \in R_q^{(m-k)\times k}$, invertible tag $h \in R_q$, a syndrome $u \in R_q$, and Gaussian parameters ζ, α, σ.
- **Ouput:** A vector $\mathbf{x}$ follows $D_{\Lambda_q^u(\mathbf{a}),\zeta}$.
- **Execute:**
  1) Choose $\mathbf{p} \leftarrow \mathsf{SampleP}(q, \zeta, \alpha, \mathbf{R})$ and set $v \leftarrow h^{-1}(u - \mathbf{a}^t\mathbf{p}) \in R_q$.
  2) Choose $\mathbf{z} \leftarrow \mathsf{SamplePolyG}(\sigma, v) \in R^k$.
  3) return $\mathbf{x} \leftarrow \mathbf{p} + \left[\begin{smallmatrix}\mathbf{R}\\ \mathbf{I}_k\end{smallmatrix}\right]\mathbf{z}$.
- $\mathsf{SampleP}(q, \zeta, \alpha, \mathbf{R}) \to \mathbf{p}$: On input a ring modulus q, Gaussian parameters ζ, α and $\mathbf{R} \in R_q^{(m-k)\times k}$, outputs $\mathbf{p}$ from $D_{R^m,\sqrt{\Sigma_\mathbf{p}}}$, where $\Sigma_\mathbf{p} = \zeta^2\mathbf{I}_m - \alpha^2\left[\begin{smallmatrix}\mathbf{R}\\ \mathbf{I}_k\end{smallmatrix}\right][\mathbf{R}^t\ \mathbf{I}_k]$, with $\zeta > s_1(\mathbf{R})\alpha$.
- $\mathsf{SamplePolyG}(\sigma, v) \to \mathbf{z}$: On input a Gaussian parameter σ and a target $v \in R_q$, outputs $\mathbf{z}$ from $D_{\Lambda_q^\perp(\mathbf{g}^t),\alpha,v}$ with $\alpha = \sqrt{5}\sigma$.

$\mathsf{DelTrap}((\mathbf{a}, \mathbf{a}_1), \mathbf{R}, h', \sigma')$:

- **Input:** A vector of polynomials $\mathbf{a}' = (\mathbf{a}, \mathbf{a}_1) \in R_q^m \times R_q^k$, **g**-Trapdoor $\mathbf{R} \in R^{(m-k)\times k}$ for $\mathbf{a}$, an invertible $h' \in R_q$ and Gaussian parameter σ'.
- **Ouput:** A **g**-trapdoor $\mathbf{R}' \in R^{m\times k}$ for $\mathbf{a}'$ with tag h'.
- **Execute:**
  1) Using SamplePre with Gaussian parameter σ' to sample each column of $\mathbf{R}'$ such that $\mathbf{a}^t\mathbf{R}' = h'\mathbf{g}^t - \mathbf{a}_1^t \bmod q$.

In what follows, for any $\mathbf{A}$ we use the notation $\mathbf{A} \in R_q^{1\times m}$ meaning that $\mathbf{A}$ is a row vector of m polynomials. In contrast, $\mathbf{A} \in R_q^{m\times 1}$ means that $\mathbf{A}$ is a column vector of m polynomials. Our ring-based scheme rIdLRS is similar to the one in Section III consisting of algorithms rIdLRS.Setup, rIdLRS.Extract, rIdLRS.Sign, rIdLRS.Verify and rIdLRS.Link working as follows:

$\mathsf{rIdLRS.Setup}(1^n)$: On input a security parameter n, do the following:

1) Choose integers $q \geq 2$, $k := \lceil \log q \rceil$, $m > k, w \geq 3$, and $M \leq 3$ fixed.
2) Choose $\sigma_1, \sigma_2, \sigma_3, \sigma$ to be Gaussian parameters.
3) Choose three hash functions $H_1 : \{0,1\}^* \to R_q^{1\times k}$, $H_2 : \{0,1\}^* \to S_{n,w}$, where $S_{n,w} := \{c \in \{0,1\}[x]/ : \deg(c) < n, \|c\|_1 = w\} \subseteq R_q$, and $H_3 : \{0,1\}^* \to R_q^{1\times(m+k)}$.
4) Run $\mathsf{GenTrap}(m, q, \sigma_1, h = 1)$ to get $\mathbf{A} \in R_q^{1\times m}$ along with a **g**-trapdoor $\mathbf{R} \in R^{(m-k)\times k}$ via D_{σ_1}.
5) The public key is $pk := \mathbf{A}$ and the master secret key is $msk := \mathbf{R}$ and system public parameter pp consists of H_1, H_2, H_3 and the rest parameters.

$\mathsf{rIdLRS.Extract}(id_i, msk)$: On input an identity $id_i \in \{0,1\}^*$ of a user in a ring and a master secret key $msk = \mathbf{R}$, do:

1) Compute $\mathbf{Q}_i = H_1(id_i) \in R_q^{1\times k}$ and let $\mathbf{A}_i := [\mathbf{A}|\mathbf{Q}_i] \in R_q^{1\times(m+k)}$.
2) Sample $\mathbf{R}_i \in R^{m\times k} \leftarrow \mathsf{DelTrap}(\mathbf{A}, \mathbf{Q}_i, \mathbf{R}, h = 1, \sigma_2)$,via D_{σ_2}.
3) Sample $\mathbf{S}_i \in R^{(m+k)\times 1} \leftarrow \mathsf{SamplePre}(\mathbf{A}_i, \mathbf{R}_i, h = 1, q, \zeta, \alpha, \sigma_3)$ such that $\mathbf{A}_i\mathbf{S}_i = q \bmod q$, hence $\mathbf{A}_i\mathbf{S}_i = q \bmod 2q$.
4) Output the private key $sk_{id_i} := \mathbf{S}_i$.

$\mathsf{rIdLRS.Sign}(\mu, event, \mathcal{R}, sk_s)$: On input a message μ, an event $event$, a ring of ℓ users $\mathcal{R} = (id_1, ..., id_\ell)$, an identity $id_s \in \mathcal{R}$ and a corresponding key $sk_s = \mathbf{S}_s$, do:

1) Let $\mathbf{K} := H_3(event) \in R_q^{1\times(m+k)}$ and $e := \mathbf{K}\mathbf{S}_s \in R_q$.
2) Let $\widehat{\mathbf{K}} \leftarrow [2\mathbf{K}| - 2e + q] \in R_{2q}^{1\times(m+k+1)}$ and $\widehat{\mathbf{S}}_s \leftarrow \left[\begin{smallmatrix}\mathbf{S}_s\\ \mathbf{I}_n\end{smallmatrix}\right] \in R^{(m+k+1)\times 1}$. Note that $\widehat{\mathbf{K}} \cdot \widehat{\mathbf{S}}_s = q \bmod 2q$.
3) For $i \in [\ell]$, let $\widehat{\mathbf{A}}_i \leftarrow [\mathbf{A}_i|\mathbf{0}] \in R_{2q}^{1\times(m+k+1)}$.
4) Choose a vector $\mathbf{Y} \in R^{(m+k+1)\times 1}$ via D_σ.
5) Calculate $c_{s+1} = H_2(\widehat{\mathbf{A}}_s\mathbf{Y} \bmod 2q, \widehat{\mathbf{K}}\mathbf{Y} \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, e)$.
6) For each identity $id_j \in \mathcal{R} \setminus \{id_s\}$, choose a vector $\mathbf{Z}_j \in R^{(m+k+1)\times 1}$ via D_σ.
7) For $i = s+1, \cdots, \ell-1, 0, 1, \cdots, s-1$, do:
   - Calculate $c_{i+1} = H_2(\widehat{\mathbf{A}}_i\mathbf{Z}_i + qc_i \bmod 2q, \widehat{\mathbf{K}}\mathbf{Z}_i + qc_i \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, e)$.
8) For $j = s$, choose $b \xleftarrow{\$} \{0,1\}$ and calculate $\mathbf{Z}_s \leftarrow (-1)^b\widehat{\mathbf{S}}_sc_s + \mathbf{Y} \bmod 2q$ and output $\mathbf{Z}_s$ with probability $\min\left\{\frac{D_\sigma^{(m+k+1)\times 1}(\mathbf{Z}_s)}{M\cdot D^{(m+k+1)\times 1}_{(-1)^b\widehat{\mathbf{S}}_sc_s,\sigma}(\mathbf{Z}_s)}, 1\right\}$
9) Output the ring signature $\sigma_\mathcal{R} = \sigma_\mathcal{R}(\mu, event) = \left(\{\mathbf{Z}_j\}_{j\in[\ell]}, c_1, \mathcal{R}, e\right)$.

$\mathsf{rIdLRS.Verify}(\mu, event, \sigma_\mathcal{R})$: Take as input a message μ, an event $event$ and a signature $\sigma_\mathcal{R} = \left(\{\mathbf{Z}_j\}_{j\in[\ell]}, c_1, \mathcal{R}, e\right)$, do the following:

1) If for all $j \in [\ell]$, $\|\mathbf{Z}_j\|_\infty \leq \eta\sigma$ where $1.1 \leq \eta \leq 1.3$ go to Step 2; otherwise output 0.
2) Let $\mathbf{K} = H_3(event)$, and let $\widehat{\mathbf{K}} \leftarrow [\mathbf{K}| - e + q] \in R_{2q}^{1\times(m+k+1)}$.
3) For $i \in [\ell - 1]$, do:
   - Let $\widehat{\mathbf{A}}_i \leftarrow [\mathbf{A}_i|\mathbf{0}] \in R_{2q}^{1\times(m+k+1)}$,
   - Calculate $c_{i+1} = H_2(\widehat{\mathbf{A}}_i\mathbf{Z}_i + qc_i \bmod 2q, \widehat{\mathbf{K}}\mathbf{Z}_i + qc_i \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, e)$.

4) If $c_1 = H_2(\widehat{\mathbf{A}}_\ell \mathbf{Z}_\ell + qc_\ell, \widehat{\mathbf{K}}\mathbf{Z}_\ell + qc_\ell \bmod 2q, \widehat{\mathbf{K}}, \mathcal{R}, \mu, event, e)$ output 1, otherwise output 0.

rIdLRS.Link($\sigma_{\mathcal{R}_1}, \sigma_{\mathcal{R}_2}$): Take as input two sing signatures ($\sigma_{\mathcal{R}_1} = (\{\mathbf{Z}_{1,j}\}_{j\in[\ell]}, c_{1,1}, \mathcal{R}_1, e_1)$) and ($\sigma_{\mathcal{R}_2} = (\{\mathbf{Z}_{2,j}\}_{j\in[\ell]}, c_{1,2}, \mathcal{R}_2, e_2)$), perform:

1) Output linked if both $\sigma_{\mathcal{R}_1}$ and $\sigma_{\mathcal{R}_2}$ are valid and $e_1 = e_2$. Otherwise output unlinked.

1) Correctness and Security.

The correctness and security of rIdLRS are proved in the same way as in Section IV. That is, we have rIdLRS is anonymity, unforgeability, linkablility ad nonslanderability under the hardness of ring-SIS$_{m,q,2\eta\sigma}$ problem, with $1.1 \le \eta \le 1.3$.

2) Parameters and Sizes for the ring version.

Basically, setting paremeters for the ring-SIS based version is similar to that for the SIS-based version but with some care. More specifically,

- n, a power of 2, is the exponent of the cyclotomic polynomial $x^n + 1$.
- q is a prime such that $q = 1 \mod (2n)$.
- For GenTrap to work: $k = \lceil \log q \rceil, m - k > 1$. Following [9], we can choose $m - k = 2$.
- For Gaussian parameter in GenTrap: Via [33], for any $\epsilon = \text{negl}(n)$, we should choose $\sigma_1 \ge \eta_\epsilon(\mathbb{Z})$, i.e., $\sigma_1 = \sqrt{\frac{\ln(2(1+1/\epsilon))}{\pi}}$.
- For Gaussian parameter in $\mathbf{R}_i \in R^{m\times k} \leftarrow$ DelTrap($\mathbf{A}, \mathbf{Q}_i, \mathbf{R}, \sigma_2$): By Lemma 13, $s_1(\mathbf{R}) \le \sigma_1\sqrt{n} \cdot O(\sqrt{m-k} + \sqrt{k} + \omega(\sqrt{\log n}))$. We need $\sigma_2 \ge \eta_\epsilon(\Lambda_q^\perp(\mathbf{A}))$, i.e., we choose $\sigma_2 \ge \sqrt{5}\cdot(s_1(\mathbf{R}) + 1)\cdot\omega(\sqrt{\log n})$. Note that, by Lemma 13, $s_1(\mathbf{R}_i) \le \sigma_2 \cdot \sqrt{n} \cdot O(\sqrt{m} + \sqrt{k} + \omega(\sqrt{\log n})$.
- For Gaussian parameter in $\mathbf{S}_i \in R^{(m+k)\times 1} \leftarrow$ SamplePre($\mathbf{A}_i, \mathbf{R}_i, h = 1, q, \zeta, \alpha, \sigma_3$): $\sigma_3 \ge \sqrt{7\cdot(s_1(\mathbf{R}_i)^2+1)}\cdot\omega(\sqrt{\log n})$, $\alpha = \sqrt{5}\sigma_3$, $\zeta > s_1(\mathbf{R}_i)\alpha$.
- Gaussian parameter in rejection sampling: Byy Lemma 13, we have $s_1(\mathbf{S}_s) \le \sigma\sqrt{n}\cdot O(\sqrt{m+k} + \omega(\sqrt{\log n}))$. By Item 4 of Lemma 4, we choose $\sigma \ge \omega(\|(-1)^b\widehat{\mathbf{S}}_s c_s\| \cdot \sqrt{\log(m+k+1)n})$. Hence, we can choose $\sigma \ge \omega(s_1(\mathbf{S}_s)\cdot\|c_s\|\cdot\sqrt{\log(m+k+1)n})$. Then, we should choose $\sigma \ge \omega(\sigma_3\sqrt{n}(\sqrt{m+k} + \sqrt{\log n})\cdot w \cdot \sqrt{\log(m+k+1)n})$.
- For the ring-SIS$_{m+k,q,\beta}$ problem to be hard: $q > 2\theta\beta(m+k)n^{1.5}\log n$ where $\beta = 2\eta\sigma$ with $1.1 \le \eta \le 1.3$, and $\theta := \max_{g\in\mathbb{Z}[x], \deg(g)\le 3(n-1)} \frac{\|g \bmod (x^n+1)\|_\infty}{\|g\|_\infty}$.
- Choose M in the rejection sampling: $M \approx e^{1+1/288} \le 3$ as in Section III-B.

VII. CONCLUSION

In this paper, we present the first (integer and ideal) lattice-based construction of identity-based linkable ring signature. We prove that the IdLRS construction enjoys the anonymity, unforgeability and nonslanderability properties in the random oracle model basing on the hardness of SIS and ring-SIS problems. As a proof of concept, we also do implementation and run some experiments to evaluate the running times of the algorithms in the proposed IdLRS and the sizes of keys and the size of signature. An efficient lattice-based IdLRS construction without a random oracle model will be an attractive research topic for future work.

REFERENCES

[1] W. Alberto Torres, V. Kuchta, R. Steinfeld, A. Sakzad, J. K. Liu, and J. Cheng. Lattice RingCT V2.0 with Multiple Input and Multiple Output Wallet. In J. Jang-Jaccard and F. Guo, editors, Information Security and Privacy, pages 156–175, Cham, 2019. Springer International Publishing.

[2] W. A. Alberto Torres, R. Steinfeld, A. Sakzad, J. K. Liu, V. Kuchta, N. Bhattacharjee, M. H. Au, and J. Cheng. Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice ringct v1.0). In W. Susilo and G. Yang, editors, Information Security and Privacy, pages 558–576, Cham, 2018. Springer International Publishing.

[3] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen. Constant-size id-based linkable and revocable-iff-linked ring signature. In R. Barua and T. Lange, editors, Progress in Cryptology - INDOCRYPT 2006, pages 364–378, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[4] M. H. Au, J. K. Liu, W. Susilo, and T. H. Yuen. Secure id-based linkable and revocable-iff-linked ring signature with constant-size construction. Theoretical Computer Science, 469:1 – 14, 2013.

[5] M. H. Au, W. Susilo, and S.-M. Yiu. Event-oriented k-times revocable-iff-linked group signatures. In L. M. Batten and R. Safavi-Naini, editors, Information Security and Privacy, pages 223–234, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[6] S. Barber, X. Boyen, E. Shi, and E. Uzun. Bitter to Better — How to Make Bitcoin a Better Currency. In A. D. Keromytis, editor, Financial Cryptography and Data Security, pages 399–414, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[7] C. Baum, H. Lin, and S. Oechsner. Towards practical lattice-based one-time linkable ring signatures. In D. Naccache, S. Xu, S. Qing, P. Samarati, G. Blanc, R. Lu, Z. Zhang, and A. Meddahi, editors, Information and Communications Security, pages 303–322, Cham, 2018. Springer International Publishing.

[8] M. Bellare and G. Neven. New Multi-Signature Schemes and a General Forking Lemma. Full version, available from, 2006.

[9] P. Bert, P.-A. Fouque, A. Roux-Langlois, and M. Sabt. Practical implementation of ring-sis/lwe based signature and ibe. In T. Lange and R. Steinwandt, editors, Post-Quantum Cryptography, pages 271–291, Cham, 2018. Springer International Publishing.

[10] X. Boyen and T. Haines. Forward-Secure Linkable Ring Signatures. In W. Susilo and G. Yang, editors, Information Security and Privacy, pages 245–264, Cham, 2018. Springer International Publishing.

[11] V. Buterin. A next-generation smart contract and decentralized application platform. 2014.

[12] S. S. M. Chow, W. Susilo, and T. H. Yuen. Escrowed linkability of ring signatures and its applications. In P. Q. Nguyen, editor, Progress in Cryptology - VIETCRYPT 2006, pages 175–192, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[13] L. Deng, Y. Jiang, and B. Ning. Identity-Based Linkable Ring Signature Scheme. IEEE Access, 7:153969–153976, 2019.

[14] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice Signatures and Bimodal Gaussians. In R. Canetti and J. A. Garay, editors, Advances in Cryptology – CRYPTO 2013, pages 40–56, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[15] L. Ducas and D. Micciancio. Improved Short Lattice Signatures in the Standard Model. In J. A. Garay and R. Gennaro, editors, Advances in Cryptology – CRYPTO 2014, pages 335–352, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[16] B. Forum. GHash.IO and double-spending against BetCoin Dice. Accessed on 23 July, 2020, 2013. [Online] Available:https://bitcointalk.org/index.php?topic=327767.0.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3087808, IEEE Access

[17] M. Gagné. Identity-Based Encryption, pages 280–282. Springer US, Boston, MA, 2005.

[18] N. Genise and D. Micciancio. Faster gaussian sampling for trapdoor lattices with arbitrary modulus. In J. B. Nielsen and V. Rijmen, editors, Advances in Cryptology – EUROCRYPT 2018, pages 174–203, Cham, 2018. Springer International Publishing.

[19] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, STOC '08, pages 197–206, New York, NY, USA, 2008. ACM.

[20] B. Holmes. e-voting: the promise and the practice. Accessed on 23 July, 2020, 2012. [Online] Available:https://parlinfo.aph.gov.au/parlInfo/download/library/prspub/1979752/upload_binary/1979752.pdf;fileType=application/pdf.

[21] i. Jeong, J. Kwon, and D. Lee. Analysis of revocable-iff-linked ring signature scheme. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E92-A(1):322–325, Jan. 2009.

[22] R. W. F. Lai, H. K. F. Cheung, and S. S. M. Chow. Trapdoors for Ideal Lattices with Applications. In D. Lin, M. Yung, and J. Zhou, editors, Information Security and Cryptology, pages 239–256, Cham, 2015. Springer International Publishing.

[23] H. Q. Le, D. H. Duong, and W. Susilo. A Blind Ring Signature Based on the Short Integer Solution Problem. In I. You, editor, Information Security Applications, pages 92–111, Cham, 2020. Springer International Publishing.

[24] W. Li, Y. Wang, L. Chen, X. Lai, X. Zhang, and J. Xin. A Simpler and Modular Construction of Linkable Ring Signature. Cryptology ePrint Archive, Report 2020/333, 2020. https://eprint.iacr.org/2020/333.

[25] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou. Linkable Ring Signature with Unconditional Anonymity. IEEE Transactions on Knowledge and Data Engineering, 26(1):157–165, 2014.

[26] J. K. Liu, V. K. Wei, and D. S. Wong. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups. In H. Wang, J. Pieprzyk, and V. Varadharajan, editors, Information Security and Privacy, pages 325–335, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[27] J. K. Liu and D. S. Wong. Linkable ring signatures: Security models and new schemes. In O. Gervasi, M. L. Gavrilova, V. Kumar, A. Laganà, H. P. Lee, Y. Mun, D. Taniar, and C. J. K. Tan, editors, Computational Science and Its Applications – ICCSA 2005, pages 614–623, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[28] X. Lu, M. H. Au, and Z. Zhang. Raptor: A practical lattice-based (linkable) ring signature. In R. H. Deng, V. Gauthier-Umaña, M. Ochoa, and M. Yung, editors, Applied Cryptography and Network Security, pages 110–130, Cham, 2019. Springer International Publishing.

[29] V. Lyubashevsky. Lattice Signatures Without Trapdoors. Cryptology ePrint Archive, Report 2011/537, Full version of paper appearing at Eurocrypt 2012, last revised 18 Oct 2017, 2012.

[30] V. Lyubashevsky. Digital signatures based on the hardness of ideal lattice problems in all rings. Cryptology ePrint Archive, Report 2016/796, 2016. https://eprint.iacr.org/2016/796.

[31] V. Lyubashevsky and D. Micciancio. Generalized Compact Knapsacks Are Collision Resistant. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, Automata, Languages and Programming, pages 144–155, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[32] V. Lyubashevsky, C. Peikert, and O. Regev. A toolkit for ring-lwe cryptography. In Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings, pages 35–54, 2013.

[33] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings, pages 700–718, 2012.

[34] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. SIAM Journal on Computing, 37(1):267–302, 2007.

[35] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[36] S. Noether. Ring Signature Confidential Transactions for Monero. Cryptology ePrint Archive, Report 2015/1098, 2015. https://eprint.iacr.org/2015/1098.

[37] S. Noether, A. Mackenzie, and T. Lab. Ring Confidential Transactions. Ledger, 1:1–18, 12 2016.

[38] M. Ober, S. Katzenbeisser, and K. Hamacher. Structure and anonymity of the bitcoin transaction graph. Future Internet, 5:237–250, 06 2013.

[39] C. Peikert. An efficient and parallel gaussian sampler for lattices. In T. Rabin, editor, Advances in Cryptology – CRYPTO 2010, pages 80–97, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[40] C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In S. Halevi and T. Rabin, editors, Theory of Cryptography, pages 145–166, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[41] R. L. Rivest, A. Shamir, and Y. Tauman. How to Leak a Secret. In C. Boyd, editor, Advances in Cryptology — ASIACRYPT 2001, pages 552–565, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[42] D. Ron and A. Shamir. Quantitative analysis of the full bitcoin transaction graph. In A.-R. Sadeghi, editor, Financial Cryptography and Data Security, pages 6–24, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[43] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In G. R. Blakley and D. Chaum, editors, Advances in Cryptology, pages 47–53, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg.

[44] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings 35th Annual Symposium on Foundations of Computer Science, pages 124–134, Nov 1994.

[45] S.-F. Sun, M. H. Au, J. K. Liu, and T. H. Yuen. Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In S. N. Foley, D. Gollmann, and E. Snekkenes, editors, Computer Security – ESORICS 2017, pages 456–474, Cham, 2017. Springer International Publishing.

[46] P. P. Tsang and V. K. Wei. Short linkable ring signatures for e-voting, e-cash and attestation. In R. H. Deng, F. Bao, H. Pang, and J. Zhou, editors, Information Security Practice and Experience, pages 48–60, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[47] P. P. Tsang, V. K. Wei, T. K. Chan, M. H. Au, J. K. Liu, and D. S. Wong. Separable linkable threshold ring signatures. In A. Canteaut and K. Viswanathan, editors, Progress in Cryptology - INDOCRYPT 2004, pages 384–398, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[48] X. Wang, Y. Chen, and X. Ma. Generic construction of linkable ring signature. IACR Cryptol. ePrint Arch., 2019:371, 2019.

[49] Q. Ye, W. Wang, Y. Tang, X. Yan, J. Zhang, Z. Zhao, and P. Qin. RLWE Commitment-Based Linkable Ring Signature Scheme and Its Application in Blockchain. In Z. Zheng, H.-N. Dai, M. Tang, and X. Chen, editors, Blockchain and Trustworthy Systems, pages 15–32, Singapore, 2020. Springer Singapore.

[50] T. H. Yuen, S. feng Sun, J. K. Liu, M. H. Au, M. F. Esgin, Q. Zhang, and D. Gu. Ringct 3.0 for blockchain confidential transaction: Shorter size and stronger security. IACR Cryptol. ePrint Arch., 2019:508, 2019.

[51] H. Zhang, F. Zhang, H. Tian, and M. H. Au. Anonymous Post-Quantum Cryptocash. In S. Meiklejohn and K. Sako, editors, Financial Cryptography and Data Security, pages 461–479, Berlin, Heidelberg, 2018. Springer Berlin Heidelberg.

[52] G. Zhao and M. Tian. A Simpler Construction of Identity-Based Ring Signatures from Lattices. In J. Baek, W. Susilo, and J. Kim, editors, Provable Security, pages 277–291, Cham, 2018. Springer International Publishing.